J Sci Comput (2009) 38: 251–289
DOI 10.1007/s10915-008-9239-z

REVIEW ARTICLE

High Order Strong Stability Preserving Time Discretizations

Sigal Gottlieb · David I. Ketcheson · Chi-Wang Shu

Received: 18 April 2008 / Revised: 5 August 2008 / Accepted: 6 August 2008 /
Published online: 5 September 2008

Abstract Strong stability preserving (SSP) high order time discretizations were developed to ensure nonlinear stability properties necessary in the numerical solution of hyperbolic partial differential equations with discontinuous solutions. SSP methods preserve the strong stability properties—in any norm, seminorm or convex functional—of the spatial discretization coupled with first order Euler time stepping. This paper describes the development of SSP methods and the connections between the timestep restrictions for strong stability preservation and contractivity. Numerical examples demonstrate that common linearly stable but not strong stability preserving time discretizations may lead to violation of important boundedness properties, whereas SSP methods guarantee the desired properties provided only that these properties are satisfied with forward Euler timestepping. We review optimal explicit and implicit SSP Runge–Kutta and multistep methods, for linear and nonlinear problems. We also discuss the SSP properties of spectral deferred correction methods.

Keywords Strong stability preserving · Runge–Kutta methods · Multistep methods · Spectral deferred correction methods · High order accuracy · Time discretization

The work of S. Gottlieb was supported by AFOSR grant number FA9550-06-1-0255.
The work of D.I. Ketcheson was supported by a US Dept. of Energy Computational Science Graduate Fellowship under grant DE-FG02-97ER25308.
The research of C.-W. Shu is supported in part by NSF grants DMS-0510345 and DMS-0809086.

S. Gottlieb (✉)
Department of Mathematics, University of Massachusetts Dartmouth, North Dartmouth, MA 02747, USA
e-mail: sgottlieb@umassd.edu

D.I. Ketcheson
Department of Applied Mathematics, University of Washington, Seattle, WA 98195-2420, USA
e-mail: ketch@amath.washington.edu

C.-W. Shu
Division of Applied Mathematics, Brown University, Providence, RI 02912, USA
e-mail: shu@dam.brown.edu

1 Overview

1.1 Introduction to SSP Methods

Linear stability theory is often used to prove convergence of numerical approximations to the solutions of partial differential equations (PDEs). Given a linear differential equation and consistent linear numerical method, linear stability is necessary and sufficient for convergence ([85] Theorem 1.5.1). Even for nonlinear PDEs, if a numerical method is consistent and its *linearization* is L_2 stable and adequately dissipative, then for sufficiently smooth problems the nonlinear approximation is convergent [84]. However, when dealing with discontinuous solutions of hyperbolic PDEs, linear stability theory no longer guarantees convergence. For example, the linearly stable second order Lax-Wendroff scheme for the nonlinear Burgers equation is L_2 nonlinearly unstable near stagnation points [63]. In this case, some kind of nonlinear stability is necessary in order to guarantee convergence.

Discontinuous solutions often arise in the solution of hyperbolic PDEs, such as hyperbolic conservation laws:

$$U_t + f(U)_x = 0. \tag{1.1}$$

In the method of lines approach, one first applies some spatial discretization, denoted $-F(u)$, to approximate the spatial derivative $f(U)_x$, yielding a semi-discrete system of ODEs

$$u_t = F(u), \tag{1.2}$$

where u is a vector of approximations to U, $u_j \approx U(x_j)$. (We will later use the notation u^n to be the fully discrete vector $u_j^n \approx U(t^n, x_j)$.) The spatial discretization $F(u)$ is carefully chosen to be nonlinearly stable under forward Euler integration. For hyperbolic PDEs, the relevant nonlinear stability property typically takes the form of total variation diminishing (TVD), total variation bounded (TVB), or some non-oscillatory requirement. These requirements may be desired even for a linear problem, where they are not necessarily required for convergence. While linear stability can often be studied directly even for complex time discretizations, nonlinear stability is more difficult to examine. Consequently, a tremendous amount of effort has been placed on the development of high order spatial discretizations which, when coupled with the forward Euler time stepping method, have the desired nonlinear stability properties for approximating discontinuous solutions of hyperbolic PDEs (see, e.g. [13, 31, 53, 59, 67, 87, 88]). However, for actual computation, higher order time discretizations are usually needed. There is no guarantee that a spatial discretization that is strongly stable in some desired norm or semi-norm (e.g., L_∞, or TV) for a nonlinear problem under forward Euler integration will possess the same nonlinear stability property when coupled with a linearly stable higher order time discretization. High order strong stability preserving time discretization methods guarantee that the nonlinear stability properties satisfied by the spatial discretization when coupled with forward Euler integration will be preserved when the same spatial discretization is coupled with these higher order methods. In Sect. 1.2, we show examples in which this condition is needed.

The idea behind strong stability preserving methods is to begin with a method of lines semi-discretization that is strongly stable in a certain norm, semi-norm, or convex functional under forward Euler time stepping, when the timestep Δt is suitably restricted, and then try

to find a higher order time discretization that maintains strong stability for the same norm, perhaps under a different timestep restriction. In other words, given a semi-discretization of the form (1.2) and convex functional $\|\cdot\|$, it is assumed that there exists a value Δt_{FE} such that

$$\|u^n + \Delta t F(u^n)\| \le \|u^n\| \quad \text{for } 0 \le \Delta t \le \Delta t_{\mathrm{FE}}. \tag{1.3}$$

A s-step numerical method for (1.2) computes the next solution value u^{n+1} from previous values $u^{n-s+1}, \ldots, u^n$. We say that the method is *strong stability preserving* (SSP) with SSP coefficient c if (in the solution of (1.2)) it holds that

$$\|u^{n+1}\| \le \max\left\{\|u^n\|, \|u^{n-1}\|, \ldots, \|u^{n-s+1}\|\right\}, \tag{1.4}$$

whenever (1.3) holds for timestep satisfies

$$\Delta t \le c\Delta t_{\mathrm{FE}} \quad \text{for some } c > 0. \tag{1.5}$$

In the case of a one-step method (1.4) reduces to

$$\|u^{n+1}\| \le \|u^n\|. \tag{1.6}$$

The class of high order SSP time discretization methods was first developed in [79] and [77] and called TVD (total variation diminishing) time discretizations. This class of methods was further studied in [26, 28, 29, 38, 40, 45, 47, 60, 72, 74, 75, 78, 81–83]. SSP methods preserve the nonlinear stability properties of forward Euler in any norm or semi-norm. In fact, since the stability arguments are based on convex decompositions of high-order methods in terms of the first-order Euler method, any convex function (such as the cell entropy stability property of high order schemes studied in [66, 68]) will be preserved by SSP high-order time discretizations.

When the timestep is limited by a linear stability requirement, or even by a nonlinear stability requirement involving an inner-product norm, there exist some well-known classes of implicit methods that allow the use of arbitrarily large timesteps. One might then hope for implicit methods that are unconditionally stable in the SSP sense. Indeed, if the spatial discretization is strongly stable in some norm under forward Euler time integration, then the fully discrete solution will also be strongly stable, in the same norm, for the implicit Euler method, without any timestep restriction [32, 40]. However, for both Runge–Kutta and linear multistep methods (and in fact, for any general linear method) of order greater than one, strong stability preservation is guaranteed only under some finite timestep [80]. Furthermore, numerical searches indicate that the timestep restrictions for implicit SSP methods are not dramatically larger than those for explicit methods [23, 47, 57].

The search for optimal SSP methods has been aided by the discovery of connections between SSP theory and contractivity theory [20, 22, 32, 33]. This discovery has led to the development of new optimal and efficient SSP methods [23, 45, 47].

SSP methods are widely used in the solution of hyperbolic PDEs. They have been employed in a variety of application areas, including compressible flow [92], incompressible flow [70], viscous flow [86], two-phase flow [3, 6], relativistic flow [1, 16, 96], cosmological hydrodynamics [19], magnetohydrodynamics [2], radiation hydrodynamics [64], two-species plasma flow [54], atmospheric transport [10], large-eddy simulation [69], Maxwell's equations [11], semiconductor devices [7], lithotripsy [89], geometrical optics [12], Schrodinger equations [9, 43], and combined with a range of spatial discretizations, including discontinuous Galerkin methods [11], level set methods [6, 9, 12, 18, 43, 71],

ENO methods [1, 6, 16], WENO methods [2, 3, 7, 19, 54, 69, 89, 96], spectral finite volume methods [10, 86], and spectral difference methods [92, 93]. This list of references is inevitably only a sample. Note that all the references above involve the use of SSP Runge–Kutta methods; we are not aware of extensive use of SSP multistep methods in applications.

In this paper, we give numerical examples that demonstrate the practical relevance of SSP methods, describe the equivalence between the Shu-Osher SSP theory and contractivity theory, and collect the main results and the most useful and efficient SSP methods. The paper is organized as follows: The need for the SSP condition, classical SSP theory and its connections to contractivity will be described in Sect. 1, together with some related stability concepts. Section 2 describes order barriers and timestep restrictions arising from contractivity theory for Runge–Kutta, linear multistep, and general linear methods. The best known methods in terms of computational efficiency for both explicit and implicit Runge–Kutta and multistep methods are presented in Sects. 3 and 4, respectively. In Sect. 5 we address the SSP property as applied to the spectral deferred correction methods and their connection to Runge–Kutta methods. Our conclusions appear in Sect. 6.

1.2 The Need for SSP Methods

When numerically solving an equation of the form

$$U_t + f(U)_x = 0$$

by the method of lines, it is important to consider the properties of the spatial discretization combined with the time discretization. If the problem is smooth, it is sufficient to linearize the problem and analyze the L_2 stability properties of the resulting discretization. However, if the solution is nonsmooth, stability in the L_2 norm is not sufficient. This is because for PDEs with discontinuous solutions, the presence of oscillations prevents the approximation from converging uniformly. To ensure that the method does not allow oscillations to form, we require stability in the maximum norm or in the TV semi-norm.

To prove stability of modern numerical methods for nonlinear hyperbolic problems with discontinuous solutions, one must typically analyze a highly nonlinear, complex spatial discretization combined with a high order time discretization. While this may be done numerically for several test cases of interest (see [49], for instance), performing a true general analysis of the stability properties of such a pair may be untenable. This is particularly the case if the stability bound of interest involves the maximum norm, or the total variation semi-norm defined by

$$\|u\|_{TV} = \sum_{j=0}^{N} |u_{j+1} - u_j|.$$

SSP time-discretizations were created to deal with problems that have these particularly challenging features (i.e., nonlinear problems, and methods, with discontinuous solutions)—for which linear stability theory is not sufficient—and which require stability in norms or semi-norms not generated by an inner product. SSP methods have proven particularly useful for integrating discontinuous Galerkin semi-discretizations, for which proofs of the TVD property are generated for each limiter used, when the method is coupled with the first order forward Euler time-discretization. It is untenable to repeat these proofs for each time-discretization used. The SSP mechanism allows these proofs to be immediately extended to all SSP higher order time-discretizations.

The SSP property is a very strong requirement that guarantees strong stability (monotonicity) in arbitrary convex functionals, for arbitrary starting values and arbitrary nonlinear, nonautonomous equations, as long as forward Euler stability is satisfied. The result of this strong stability requirement is a rather stringent restriction on the timestep. In many cases, a more relaxed timestep restriction may suffice; for example, if instead of considering arbitrary convex functionals we require monotonicity in some inner-product norm. The relevant property in this case is referred to as algebraic stability for Runge–Kutta methods and as G-stability for multistep methods, and methods of higher than first order exist that are unconditionally stability preserving in this sense. If we require strong stability preservation only when integrating linear autonomous equations (i.e., a 'linear SSP' property), the timestep condition is also more relaxed, as we will see below in Sect. 1.4.1. When dealing with smooth, well-resolved problems, a weaker condition may guarantee monotonicity; conditions for the preservation of positivity for certain smooth solutions have been investigated by Horvath [35, 36]. Furthermore, the use of special starting procedures for linear multistep methods may result in a relaxed SSP condition as well (see Sect. 4). Finally, if we require only a weaker boundedness condition, a larger stepsize may be sufficient.

When none of these special cases apply, as is frequently the case for nonlinear PDEs with discontinuous solutions, we turn to the general SSP analysis to guarantee strong stability in the desired norm. In the following we consider some examples that demonstrate the need for SSP time discretizations in the solution of hyperbolic PDEs with discontinuities.

In [55] it is shown that when a second order Lax-Wendroff scheme—which is strongly stable in the L_2 norm—is applied to the linear advection equation

$$U_t + aU_x = 0 \tag{1.7}$$

with a step-function initial condition, there will be an overshoot (for $a > 0$) or undershoot (for $a < 0$) near the discontinuity. In fact, it is shown that the Gibbs phenomenon will affect *any* finite difference scheme of second (or higher) order accuracy applied to this problem. In other words, an overshoot or undershoot that prevents uniform convergence will occur for all finite difference methods of second order, even those that are strongly stable in the L_2 norm, for linear problems. Clearly, L_2 stability is not the relevant property when we desire well-behaved numerical solutions of hyperbolic PDEs with discontinuous solutions. However, if we can prevent oscillations from forming by requiring stability in the maximum norm or the TV semi-norm, we can obtain uniform convergence.

Even when the spatial discretization is total variation diminishing (TVD) when coupled with forward Euler integration, this is not sufficient to guarantee that it will be TVD when combined with a higher order time-discretization. Consider Burgers' equation

$$U_t + \left(\frac{1}{2}U^2\right)_x = 0, \quad x \in [0, 2) \tag{1.8}$$

with initial condition

$$u(0, x) = \frac{1}{2} - \frac{1}{4}\sin(\pi x) \tag{1.9}$$

and periodic boundary conditions. The solution is right-traveling and over time steepens into a shock. We discretize using the conservative upwind approximation

$$-f(U)_x \approx F(u) = -\frac{1}{\Delta x}\left(f(u_j) - f(u_{j-1})\right) \tag{1.10}$$

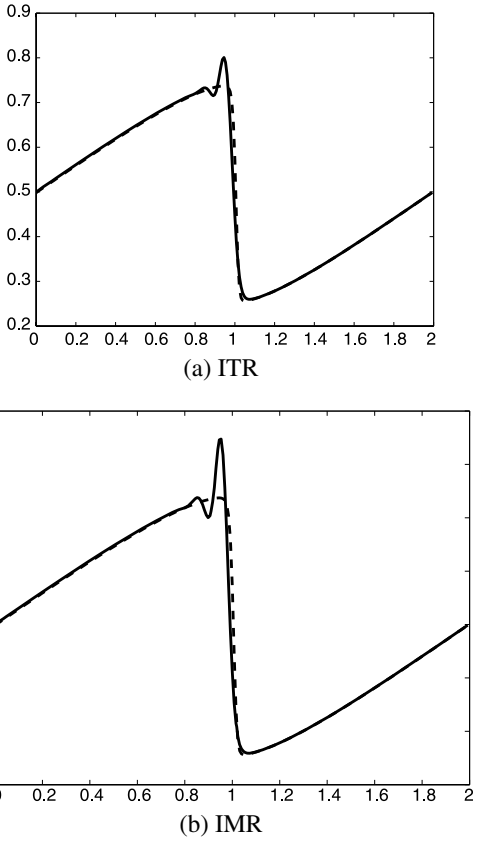


Fig. 1 Solution of Burgers' equation (1.8) at $t = 2.0$ using upwind differencing with 256 spatial points and the implicit trapezoidal rule (ITR) (1.11) (**a**) and implicit midpoint rule (IMR) (1.12) (**b**) with $\Delta t = 8\Delta x = 8\Delta t_{\mathrm{FE}}$. The solution appears smooth until the shock develops, then an oscillation develops at the trailing edge of the shock

where $f(u_j) = \frac{1}{2}u_j^2$. Using Harten's lemma [31] and the fact that for this problem $f'(u) > 0$ we conclude that this method is TVD for $\Delta t \leq \Delta x$ when coupled with forward Euler. Using this fact we can conclude that if we integrate, instead, using backward Euler, the solution will be TVD for all values of Δt. Instead, let us use the (second-order) implicit trapezoidal rule

$$u^{n+1} = u^n + \frac{1}{2}\Delta t \left(F(u^{n+1}) + F(u^n)\right) \tag{1.11}$$

which is A-stable, like the backward Euler method. Hence the full discretization is absolutely stable under any timestep. Nevertheless, we find that whenever $\Delta t > 2\Delta x$, oscillations appear (see Fig. 1a). This is also the case when we use the implicit midpoint rule

$$\begin{aligned} u^{(1)} &= u^n + \frac{\Delta t}{2} F(u^{(1)}), \\ u^{n+1} &= u^{(1)} + \frac{\Delta t}{2} F(u^{(1)}) \end{aligned} \tag{1.12}$$

which is A-stable, L-stable, and B-stable, but clearly not SSP for $\Delta t > 2\Delta x$ (see Fig. 1b). Although for many problems implicit methods can be used to avoid stability-related timestep restrictions, this is not the case when the SSP condition is of interest.

Below we will see that the conditions required for an SSP time integrator are weaker if only linear problems are considered. For instance, while explicit Runge–Kutta methods that are SSP for nonlinear problems have order of accuracy at most four, explicit SSP Runge–Kutta methods for linear problems can easily be constructed with any order of accuracy. Although one might hope that the timestep restriction associated with the linear SSP property would suffice to give reasonably good behavior in the nonlinear case, experience shows this is not true. Consider a fifth order explicit Runge–Kutta method with six stages, due to Butcher [5, p. 174]. The Butcher array of the method is

$$\begin{array}{c|cccccc}
0 & & & & & & \\
\frac{1}{4} & \frac{1}{4} & & & & & \\
\frac{1}{4} & \frac{1}{8} & \frac{1}{8} & & & & \\
\frac{1}{2} & 0 & 0 & \frac{1}{2} & & & \\
\frac{3}{4} & \frac{3}{16} & -\frac{3}{8} & \frac{3}{8} & \frac{9}{16} & & \\
1 & -\frac{3}{7} & \frac{8}{7} & \frac{6}{7} & -\frac{12}{7} & \frac{8}{7} & \\
\hline
 & \frac{7}{90} & 0 & \frac{16}{45} & \frac{2}{15} & \frac{16}{45} & \frac{7}{90}
\end{array} \tag{1.13}$$

It can be shown this method has SSP coefficient $c = 0$. However, for linear problems, we can take advantage of the linearity property to rearrange this method into the SSP form

$$\begin{aligned}
u^{(1)} &= u^n + \Delta t \frac{9}{16} F(u^n), \\
u^{(i+1)} &= u^{(i)} + \Delta t \frac{9}{16} F(u^{(i)}) \quad (1 \le i \le 5), \\
u^{n+1} &= \frac{84449}{3^{12}} u^n + \frac{313328}{5 \cdot 3^{11}} u^{(1)} + \frac{9344}{3^{10}} u^{(2)} + \frac{137216}{3^{12}} u^{(3)} + \frac{2^{13}}{3^{11}} u^{(4)} + \frac{2^{16}}{5 \cdot 3^{12}} u^{(6)},
\end{aligned}$$

which is clearly a convex combination of forward Euler steps, and thus SSP with coefficient $c = 16/9 \approx 1.78$. (Note that when we implement the method in this form for a nonlinear problem it is still SSP with the same coefficient but it is only second order.) Indeed, this bound is seen in practice—if we apply the method to a first order upwind semi-discretization of the advection equation (1.7), the resulting solution is TVD for $\Delta t < 1.78 \Delta x$. However, the linear SSP property does not carry over to the nonlinear case. Very different behavior results when applying the method (1.13) to solve Burgers' equation (1.8) with a discontinuous initial condition. The spatial derivative is discretized using a second order TVD flux-differencing method with the superbee slope limiter. This spatial discretization is TVD when coupled with forward Euler for a timestep $\Delta t \le \frac{1}{2} \Delta x$. If we look at the linear SSP condition, we may expect that a timestep of $\Delta t \le \frac{8}{9} \Delta x$ will preserve the TVD property. However, we find that this method leads to spurious oscillations already for $\Delta t = 0.46 \Delta x$. In other words, the method fails to be TVD for even smaller timesteps than for Euler's method, whereas a purely linear SSP analysis would predict just the opposite.

The challenge of dealing with both a nonlinear method or problem and a difficult stability property arises with the weighted essentially non-oscillatory (WENO) method [42, 59]. The WENO method is an extension of the essentially non-oscillatory (ENO) method, which chooses the smoothest finite difference stencil to evaluate the derivative on. The WENO scheme uses a weighted combination of all the stencils considered, where the weights approach the center-upwind difference weights in smooth regions and approach the ENO

weights in regions near the discontinuities. The second order ENO method satisfies the TVD, hence non-oscillatory, condition when coupled with forward Euler. Even though we could not expect the same TVD property for higher order ENO methods, since TVD schemes can be at most second order accurate in space [67], we do expect similar essentially non-oscillatory performance, even though this is not rigorously enforced. We expect that since WENO weights should reduce to ENO weights near the discontinuity, WENO should behave like ENO in the region of a shock. However, there is no theory to guarantee this behavior. Despite the lack of theoretical results, in practice we observe advantages to the use of SSP methods for WENO methods on linear and nonlinear problems.

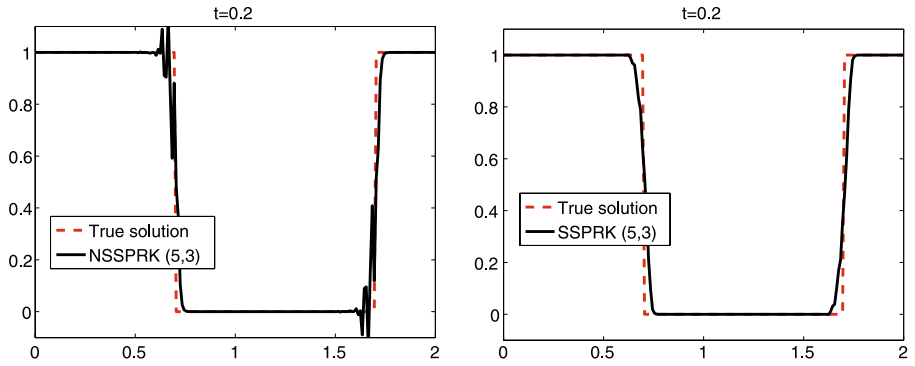
Let's consider, again, the linear advection equation (1.7), now with a square-wave initial condition:

$$U(x,0) = \begin{cases} 0 & -1 \le x \le -0.5, \\ 1 & -0.5 < x < 0.5, \\ 0 & 0.5 \le x \le 1.0, \end{cases} \tag{1.14}$$

and periodic boundary conditions. In our experiments, we use the fifth order WENO scheme of [42], denoted by WENO5, with $\epsilon = 10^{-29}$ (see [42] for the definition of ϵ) and the time discretizations SSPRK (3,3) (in (3.2)), SSPRK (10,4) (in (3.6)), the SSPRK (5,3) from [82] (see (3.3)) and the non-SSP RK method NSSPRK (5,3):

$$\begin{aligned} u^{(1)} &= u^n + \frac{1}{7}\Delta t F(u^n), \\ u^{(2)} &= u^n + \frac{3}{16}\Delta t F(u^{(1)}), \\ u^{(3)} &= u^n + \frac{1}{3}\Delta t F(u^{(2)}), \\ u^{(4)} &= u^n + \frac{2}{3}\Delta t F(u^{(3)}), \\ u^{n+1} = u^{(5)} &= u^n + \frac{1}{4}\Delta t F(u^n) + \frac{3}{4}\Delta t F(u^{(4)}), \end{aligned}$$

which was used in [91] and is L_2 linearly stable in combination with the linearized version of WENO5, obtained by setting all weights equal to the linear weights. Figure 2 shows the solution obtained with NSSPRK (5,3) at time $t = 0.2$ and using the timestep $\Delta t = 1.85\Delta x$, as suggested in [91]. We observe that at these times, oscillations are present. However, if we instead apply the SSPRK (5,3) method, no significant oscillations form. To study this example more carefully, we look at the total variation norm of the numerical solution. We are interested in the largest CFL number σ_{TVD} such that for $\Delta t \le \sigma_{\mathrm{TVD}}\Delta x$, the total variation of the solution does not increase by more than 10^{-13} at each timestep. For comparison, we also calculated the linear L_2 stability timestep restriction $\Delta t \le \sigma_{L_2}\Delta x$ of these discretizations by computing the spectrum of the underlying (fixed-coefficient) fifth order finite difference method and determining the largest timestep such that this spectrum lies within the region of absolute stability for a given method. This timestep restriction will avoid oscillations when integrating sufficiently smooth solutions. In Table 1 we present a comparison of these CFL numbers. We also compare the relative efficiency of the methods by dividing the stable CFL number by the number of function evaluations (i.e. stages, m) required for each method. We observe that in each case the timestep restriction for L_2 linear stability is larger than that required for the TVD property, and that the non-SSP method is less efficient than the SSP methods.



(a) Non-SSP (b) SSP

Fig. 2 Advection of a square-wave using a non-SSP method NSSPRK (5,3) (*left*) or SSPRK (5,3) (*right*) with $\Delta t = 1.85\Delta x$ at time $t = 0.2$

Table 1 Largest stable CFL numbers and relative efficiencies for smooth and discontinuous solutions

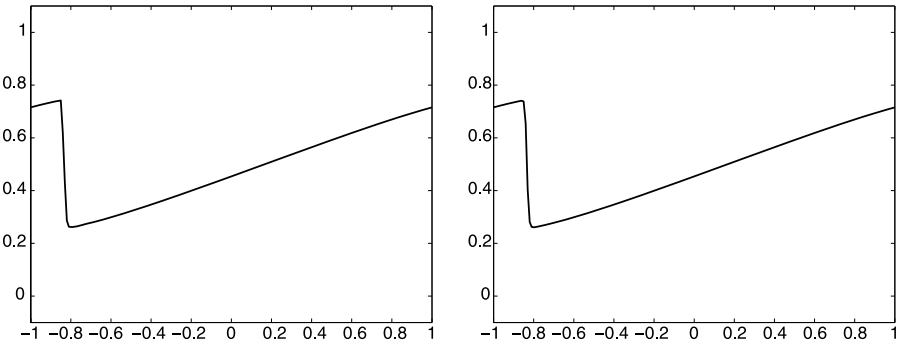
Method	σ_{L^2}	σ_{L^2}/m	σ_{TVD}	σ_{TVD}/m
SSPRK (3,3)	1.43	0.477	0.78	0.260
NSSPRK (5,3)	2.56	0.512	1.01	0.202
SSPRK (5,3)	2.04	0.408	1.31	0.262
SSPRK (10,4)	3.08	0.308	3.07	0.307

The last example demonstrates that SSP methods may reduce the computational cost when the timestep is limited by stability considerations. Next, let us compare results using different methods while fixing the computational cost. In this example we return to Burgers' equation (1.8) with initial condition (1.9), and WENO5 spatial discretization with $\epsilon = 10^{-29}$. We compare results obtained with SSPRK (3,3), SSPRK (5,3) and NSSPRK (5,3). We use 200 points in space, so that $\Delta x = 0.01$ and we choose $\Delta t = \frac{\Delta x}{.75}$ for SSPRK (3,3) and $\Delta t = \frac{5}{3}\frac{\Delta x}{.75}$ for the five stage methods (so that total number of function evaluations is the same). The solutions obtained with the SSP methods do not exhibit oscillations while the solution from the non-SSP method does, as seen in Fig. 3.

For SSP Runge–Kutta methods, it is desirable that the internal stages also be strongly stable. This means requiring not only that $\|u^{n+1}\| \leq \|u^n\|$, but also that each stage $u^{(i)}$ for $i = 1, \ldots, m$ satisfy $\|u^{(i)}\| \leq \|u^{(i-1)}\|$. Since the SSP argument relies on convexity, which is satisfied at the intermediate stages as well, SSP Runge–Kutta methods have intermediate stage SSP properties. The SSP guarantee of provable stability even for the intermediate stages is given with no additional cost.

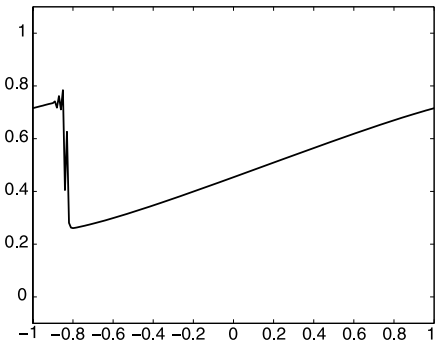
This condition is frequently necessary in the approximate solution of hyperbolic PDEs. For example, in the numerical solution of the Euler equations of gas dynamics, it is important that negative pressure or density values be avoided even in the intermediate stages. Violations of these bounds are more than theoretically problematic, as they lead to non-physical states and typically to failure of the solution algorithm. Consider the Riemann problem for the Euler equations

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ P + \rho u^2 \\ u(P+E) \end{pmatrix}_x = 0$$



(a) SSPRK (3,3)

(b) SSPRK (5,3)



(c) NSSPRK (5,3)

Fig. 3 Burgers' equation with WENO5 for different time stepping methods

where ρ is the density, ρu is the momentum, E is the energy, and $P = (\gamma - 1)(E - \frac{1}{2}\rho u^2)$ is the pressure. We take $\gamma = 1.4$, initial density and pressure equal to unity everywhere, and initial velocity:

$$v(x, 0) = \begin{cases} -3.1 & (x < 0.5), \\ 3.1 & (x > 0.5). \end{cases}$$

This leads to a near-vacuum state at $x = 0$ at later times. The solution is computed on the interval $0 \le x \le 1$ using 200 grid points. Again fifth-order WENO (without characteristic decomposition) [42] is used for the semi-discretization. We determine the largest CFL number σ for which the density and pressure values remain positive at all Runge–Kutta stages. These values are listed in Table 2, and we see that the SSP methods allow a more efficient time-step than the NSSPRK (5,3) and classical four-stage fourth order method (denoted by RK (4,4)). Clearly, the SSP methods are the most efficient for this purpose.

1.3 The Shu-Osher Formulation

Explicit SSP methods were first introduced by Shu and Osher in the following manner. First, an explicit Runge–Kutta method is written in the form [79],

$$u^{(0)} = u^n,$$

Table 2 Largest positivity-preserving CFL number for the near-vacuum Riemann problem

Method	σ	σ/m
SSPRK (3,3)	0.77	0.257
NSSPRK (5,3)	0.50	0.100
RK (4,4)	0.77	0.193
SSPRK (10,4)	2.70	0.270

$$u^{(i)} = \sum_{k=0}^{i-1} \left(\alpha_{i,k} u^{(k)} + \Delta t \beta_{i,k} F(u^{(k)})\right), \tag{1.15}$$
$$u^{n+1} = u^{(m)}.$$

The form (1.15), referred to as the Shu-Osher form, is different from the classical representation (see (1.20) below), but is more convenient for SSP analysis. The SSP property for explicit multistep methods can be analyzed based on the traditional form [77],

$$u^{n+1} = \sum_{i=1}^{m} \left(\alpha_i u^{n+1-i} + \Delta t \beta_i F(u^{n+1-i})\right). \tag{1.16}$$

Consistency requires that $\sum_{k=0}^{i-1} \alpha_{i,k} = 1$ for Runge–Kutta methods and $\sum_{k=1}^{m} \alpha_i = 1$ for multistep methods.

If all the coefficients are non-negative, the forms (1.15) and (1.16) can both easily be manipulated into convex combinations of forward Euler steps, with a modified timestep. This observation motivates the following theorem:

Theorem 1.1 ([79] Sect. 2) *If the forward Euler method applied to* (1.2) *is strongly stable under the timestep restriction* $\Delta t \le \Delta t_{\mathrm{FE}}$, i.e. (1.3) *holds, and if* $\alpha_{i,k}, \beta_{i,k} \ge 0$ ($\alpha_i, \beta_i \ge 0$ *for the multistep method*), *then the solution obtained by the Runge–Kutta method* (1.15) (*or the multistep method* (1.16)) *satisfies the strong stability bound* (1.6) (*or* (1.4)) *under the timestep restriction*

$$\Delta t \le c(\alpha, \beta) \Delta t_{\mathrm{FE}}, \tag{1.17}$$

where $c(\alpha, \beta) = \min_{i,k} \frac{\alpha_{i,k}}{\beta_{i,k}}$ (*or* $c(\alpha, \beta) = \min_i \frac{\alpha_i}{\beta_i}$ *for the multistep method*). *If any of the* β*s is equal to zero, the corresponding expression is considered infinite.*

Proof Each stage of the Runge–Kutta method (1.15) can be re-written as a convex combination of forward Euler steps:

$$\|u^{(i)}\| = \left\| \sum_{k=0}^{i-1} \left(\alpha_{i,k} u^{(k)} + \Delta t \beta_{i,k} F(u^{(k)})\right) \right\|$$
$$\le \sum_{k=0}^{i-1} \alpha_{i,k} \left\| u^{(k)} + \Delta t \frac{\beta_{i,k}}{\alpha_{i,k}} F(u^{(k)}) \right\|.$$

Now, since each $\|u^{(k)} + \Delta t \frac{\beta_{i,k}}{\alpha_{i,k}} F(u^{(k)})\| \le \|u^{(k)}\|$ as long as $\Delta t \le \frac{\beta_{i,k}}{\alpha_{i,k}} \Delta t_{\mathrm{FE}}$, and since $\sum_{k=0}^{i-1} \alpha_{i,k} = 1$ by consistency, we have $\|u^{n+1}\| \le \|u^n\|$ as long as $\Delta t \le \frac{\beta_{i,k}}{\alpha_{i,k}} \Delta t_{\mathrm{FE}}$ for all i and k.

Similarly, this result is obtained for multistep methods by using the observation that the explicit multistep method can be written as a convex combination of forward Euler steps,

$$u^{n+1} = \sum_{i=1}^{m} \left(\alpha_i u^{n+1-i} + \Delta t \beta_i F(u^{n+1-i})\right) \tag{1.18}$$

$$= \sum_{i=1}^{m} \alpha_i \left(u^{n+1-i} + \Delta t \frac{\beta_i}{\alpha_i} F(u^{n+1-i})\right), \tag{1.19}$$

with $\sum_{k=1}^{m} \alpha_i = 1$. □

Note that Theorem 1.1 gives sufficient conditions for strong stability preservation, but makes no claims about their necessity. This will be addressed in the next section. It is interesting that the stable timestep is the product of only two factors, the forward Euler timestep (Δt_{FE}), which depends on the spatial discretization alone, and the coefficient $c(\alpha, \beta)$, which depends only on the time discretization. In the literature, c has been referred to as a *CFL coefficient*. However, the CFL condition prescribes a relation between the time step and the spatial grid size, whereas the SSP coefficient describes the ratio of the strong stability preserving timestep to the strongly stable forward Euler time step.

For multistep methods, the form (1.16) is unique. For Runge–Kutta methods, a given method can be written in many ways in the Shu-Osher form (1.15). In Theorem 1.1, the coefficient $c(\alpha, \beta)$ depends on the particular Shu-Osher representation chosen. Hence it is helpful to consider the Butcher form

$$\begin{aligned} u^{(i)} &= u^n + \Delta t \sum_{j=1}^{m} a_{ij} F(u^{(j)}) \quad (1 \le i \le m), \\ u^{n+1} &= u^n + \Delta t \sum_{j=1}^{m} b_j F(u^{(j)}). \end{aligned} \tag{1.20}$$

The notation $A = (a_{ij})$ and $b = (b_j)$, allows any Runge–Kutta method given in the Butcher form to be referred to as $(\mathbf{A}, \boldsymbol{b})$. The Butcher form allows for fully implicit methods; inclusion of implicit terms in the Shu-Osher form led to an extension that we will refer to as the modified Shu-Osher form. This was first defined in [22, 33], and is given by

$$\begin{aligned} u^{(i)} &= \left(1 - \sum_{j=1}^{m} \lambda_{ij}\right) u^n + \sum_{j=1}^{m} \left(\lambda_{ij} u^{(j)} + \Delta t \mu_{ij} F(u^{(j)})\right) \quad (1 \le i \le m), \\ u^{n+1} &= \left(1 - \sum_{j=1}^{m} \lambda_{m+1,j}\right) u^n + \sum_{j=1}^{m} \left(\lambda_{m+1,j} u^{(j)} + \Delta t \mu_{m+1,j} F(u^{(j)})\right). \end{aligned} \tag{1.21}$$

We can define the term $c(\alpha, \beta)$ for the implicit case by replacing (α, β) by (λ, μ). We will use the term *SSP coefficient* and the notation $c(\mathbf{A}, \boldsymbol{b})$ (or just c) to refer to the maximal value of $c(\alpha, \beta)$ over all Shu-Osher representations of a given Runge–Kutta method, and we will see in the next section that there exists a straightforward way to determine this value.

There is a simple relation between the modified Shu-Osher representation and the Butcher representation [22, 33]. First define

$$\mathcal{L} = \begin{pmatrix} \mathcal{L}_0 \\ \mathcal{L}_1 \end{pmatrix}$$

where $\mathcal{L}_0 = (\lambda_{ij})$ for $1 \le i, j \le m$, and $\mathcal{L}_1 = (\lambda_{m+1,j})$ for $1 \le j \le m$, and

$$\mathcal{M} = \begin{pmatrix} \mathcal{M}_0 \\ \mathcal{M}_1 \end{pmatrix}$$

where $\mathcal{M}_0 = (\mu_{ij})$ for $1 \le i, j \le m$, and $\mathcal{M}_1 = (\mu_{m+1,j})$ for $1 \le j \le m$. The relation between the Shu-Osher representation and the Butcher array is

$$\mathcal{M}_0 = A - \mathcal{L}_0 A, \qquad \mathcal{M}_1 = b^T - \mathcal{L}_1 A$$

(where $I - \mathcal{L}_0$ is invertible).

1.3.1 Negative Coefficients

Up to now, we have considered only methods for which all the $\beta_{i,k}$s are nonnegative. However, the SSP property can be guaranteed also in the case where some of the $\beta_{i,k}$s are negative, provided that we modify the spatial discretization for these instances. When $\beta_{i,k}$ is negative, $\beta_{i,k} F(u^{(k)})$ is replaced by $\beta_{i,k} \tilde{F}(u^{(k)})$, where $\tilde{F}$ approximates the same spatial derivative(s) as F, but the strong stability property $\|u^{n+1}\| \le \|u^n\|$ holds for the first order Euler scheme, solved backward in time, i.e.,

$$u^{n+1} = u^n - \Delta t \tilde{F}(u^n). \tag{1.22}$$

This can be achieved, for hyperbolic conservation laws, by solving the negative in time version of (1.1),

$$U_t - f(U)_x = 0.$$

Numerically, the only difference is the change of the upwind direction. Clearly, $\tilde{F}$ can be computed with the same cost as that of computing F. Thus, if $\alpha_{i,k} \ge 0$, all the intermediate stages $u^{(i)}$ in (1.15) are simply convex combinations of backward in time Euler and forward Euler operators, with Δt replaced by $\frac{|\beta_{i,k}|}{\alpha_{i,k}} \Delta t$. Therefore, any strong stability bound satisfied by the backward in time and forward in time Euler methods will be preserved by the RK method [79].

It would seem that if both $F(u^{(k)})$ and $\tilde{F}(u^{(k)})$ must be computed for the same k, the computational cost as well as storage requirement for this k is doubled. For this reason, negative $\beta_{i,k}$ were avoided whenever possible in [26, 28, 29, 74, 83]. However, since, as shown in Proposition 3.3 of [28] and Theorem 4.1 in [74], it is not always possible to avoid negative $\beta_{i,k}$, recent studies (e.g. [27, 72, 75]) have considered efficient ways of implementing negative $\beta_{i,k}$. Firstly, inclusion of negative $\beta_{i,k}$, even when not absolutely necessary, may raise the SSP coefficient enough to compensate for the additional computational cost incurred by $\tilde{F}$. Secondly, since $\tilde{F}$ is, numerically, the downwind version of F, it is sometimes possible to compute both F and $\tilde{F}$ without doubling the computational cost [27]. Finally, if F and $\tilde{F}$ do not appear for the same k, then neither the computational cost nor the storage requirement is increased [75].

In practice, these methods are rarely used, and so in this review we deal primarily with methods with nonnegative $\beta_{i,k}$s. For more details on SSP methods with negative coefficients, see [21, 25, 27, 29, 34, 72, 75, 77, 79, 81].

1.4 Contractivity and Absolute Monotonicity

So far we have focused on the monotonicity property (1.4), which bounds the growth of solutions. Classical stability analysis focuses on bounding the growth of differences of solutions, i.e.

$$\|\boldsymbol{u}^{n+1} - \boldsymbol{v}^{n+1}\| \leq \|\boldsymbol{u}^n - \boldsymbol{v}^n\|. \tag{1.23}$$

Taking $\boldsymbol{v}$ to represent a perturbed version of $\boldsymbol{u}$ due to numerical error, we see that property (1.23) (referred to as *contractivity*) implies that errors do not grow as they are propagated. A numerical method is said to be *contractivity preserving* (or simply *contractive*) if it satisfies (1.23), possibly subject to some timestep restriction, whenever (1.23) is satisfied under forward Euler integration.

Contractivity preserving methods are closely related to absolutely monotonic functions. A function $\phi(z)$ is said to be absolutely monotonic at $z = z_0$ if $\phi(z_0)$ and all of its derivatives, $\phi^{(n)}(z_0)$, are nonnegative. The radius of absolute monotonicity of ϕ, denoted $R(\phi)$, is the largest value r such that $\phi(z)$ is absolutely monotonic for $z \in (-r, 0]$.

1.4.1 Contractivity Preservation for Linear Problems

To see this connection between absolute monotonicity and the SSP condition, consider the linear, autonomous system

$$u' = Lu, \tag{1.24}$$

where the fixed matrix L is such that the numerical solution of (1.24) is contractive, in some norm $\|\cdot\|$, under forward Euler (FE) integration, i.e.

$$\|u_n + \Delta t L u_n\| \leq \|u_n\| \quad \text{for } 0 \leq \Delta t \leq \Delta t_{\text{FE}}. \tag{1.25}$$

Note that for the solution of (1.24), the concepts of contractivity (1.23) and strong stability (1.6) are equivalent. Applying a Runge–Kutta method to (1.24) yields the iteration

$$u^{n+1} = \phi(\Delta t L) u^n \tag{1.26}$$

where ϕ is the stability function of the method. Suppose we can rewrite ϕ as a combination of iterated forward Euler steps, each of length $\Delta t_{\text{FE}} = \Delta t / R$:

$$\phi(\Delta t L) = \sum_i \omega_i \left(I + \frac{\Delta t}{R} L \right)^i, \tag{1.27}$$

and note that the coefficients ω_i must sum to one for consistency: $\phi(0) = \sum \omega_i = 1$. The monotonicity condition (1.25) implies that $\|I + \frac{\Delta t}{R} L\| \leq 1$ for $\frac{\Delta t}{R} \leq \Delta t_{\text{FE}}$, so if $\omega_i \geq 0$, it follows that $\|\phi(\Delta t L)\| \leq 1$, and hence the method is strong stability preserving.

The form (1.27) is easily obtained by expanding the stability function $\phi(z)$ in a power series about $z = -R$:

$$\phi(z) = \sum_i \frac{\phi^{(i)}(-R)}{i!} (z + R)^i = \sum_i \frac{R^i \phi^{(i)}(-R)}{i!} \left(1 + \frac{z}{R} \right)^i. \tag{1.28}$$

Comparing (1.28) with (1.27) reveals that $\omega_i = \frac{R^i \phi^{(i)}(-R)}{i!}$, so $\omega_i \geq 0$ if and only if ϕ and all its derivatives are non-negative, i.e. if $\phi(z)$ is absolutely monotonic at $z = -R$. Hence absolute monotonicity of the stability function is a sufficient condition for strong stability preservation for the linear problem (1.24); in other words, a given method is SSP for linear problems under the timestep restriction

$$\Delta t \leq R \Delta t_{\text{FE}} \tag{1.29}$$

where $R = R(\phi)$. In fact, this timestep restriction is sharp in the case that L is the first order upwind difference matrix and $\|\cdot\|$ is the maximum norm [80].

For simplicity we have considered a one-step method; applying an s-step method to (1.24) yields the iteration

$$u^{n+1} = \psi_1(\Delta t L)u^n + \psi_2(\Delta t L)u^{n-1} + \cdots + \psi_s(\Delta t L)u^{n-s+1}. \tag{1.30}$$

The method is strong stability preserving for linear constant coefficient problems (1.24) under the timestep restriction (1.29) where

$$R = \min_j R(\psi_j) \tag{1.31}$$

is known as the threshold factor [80].

Absolute monotonicity of the stability function was studied in [45, 50, 56, 57, 90] to find optimal contractive explicit and implicit methods of Runge–Kutta and linear multistep types for linear systems.

In general, since (1.24) is a special case of (1.2), we must have $c \leq R$ for any given method. For explicit linear multistep methods, $\psi_i(z) = \alpha_i + \beta_i z$, so $R(\psi_i) = \alpha_i/\beta_i$ if $\alpha_i, \beta_i \geq 0$; hence R is equal to the SSP coefficient $c(\alpha, \beta)$ from Theorem 1.1. For implicit linear multistep methods, the difference between c and R for optimal methods is known to be small [57].

1.4.2 Contractivity Preserving Multistage Methods for Nonlinear Problems

The analysis of SSP Runge–Kutta methods for nonlinear problems is more complicated. Here we briefly outline the theory; the interested reader is referred to [20, 32, 51]. In order to state the relationship between absolute monotonicity and the SSP coefficient for Runge–Kutta methods, we must restrict ourselves to *irreducible* Runge–Kutta methods—Runge–Kutta methods that are not equivalent to a method with fewer stages. For a precise definition of reducibility see, e.g., [22, Definition 3.1].

When a Runge–Kutta method is applied to a nonautonomous linear system of equations, the resulting iteration involves a generalization of the stability function, known as the *matrix-valued K-function* of the method. By considering absolute monotonicity of this function, Kraaijevanger [51] extended the concept of absolute monotonicity of a function to absolute monotonicity of a Runge–Kutta method. The radius of absolute monotonicity of a Runge–Kutta method is denoted by $R(\mathbf{A}, \boldsymbol{b})$ and is the supremum over all values of $r \geq 0$ such that

$$(I + rA)^{-1} \text{ exists} \quad \text{and}$$

$$A(I + rA)^{-1} \geq 0, \tag{1.32a}$$

$$b^T(I + rA)^{-1} \geq 0, \tag{1.32b}$$

$$rA(I + rA)^{-1} e_m \le e_m, \tag{1.32c}$$

$$rb^T (I + rA)^{-1} e_m \le 1. \tag{1.32d}$$

All inequalities are to be understood componentwise, and e_m denotes the vector of length m with all entries equal to one. If $A \ge 0, b \ge 0$ do not hold, we define $R(\mathbf{A}, \boldsymbol{b}) = 0$. Kraaijevanger showed that, for general nonlinear, nonautonomous problems, an irreducible Runge–Kutta method preserves contractivity for timesteps

$$\Delta t \le R(\mathbf{A}, \boldsymbol{b}) \Delta t_{\mathrm{FE}}$$

where Δt_{FE} is the largest contractivity preserving timestep under forward Euler integration. In fact, this restriction is sharp in the sense that one can always find some system of ODEs for which contractivity is violated when the time-step taken exceeds the threshold value.

More recently it has been proved that the radius of absolute monotonicity $R(\mathbf{A}, \boldsymbol{b})$ is equal to the optimal SSP coefficient $c(\mathbf{A}, \boldsymbol{b})$, which is the largest value of $c(\alpha, \beta)$ over all Shu-Osher forms (α, β) corresponding to the (fixed) Butcher form $(\mathbf{A}, \boldsymbol{b})$, [22, 33]. Furthermore, there is an explicit construction of a Shu-Osher form such that the optimal SSP coefficient is evident [22, 33]. If a method $(\mathbf{A}, \boldsymbol{b})$ has radius of absolute monotonicity $R(\mathbf{A}, \boldsymbol{b}) \ge 0$, we can construct the optimal Shu-Osher representation as follows:

$$\mathcal{M}_0 = A(I + cA)^{-1}, \qquad \mathcal{M}_1 = b^T (I + cA)^{-1}, \qquad \mathcal{L} = c\mathcal{M}, \quad c = R(\mathbf{A}, \boldsymbol{b})$$

for $0 \le R(\mathbf{A}, \boldsymbol{b}) < \infty$. If $R(\mathbf{A}, \boldsymbol{b}) = \infty$ we use:

$$\mathcal{L}_0 = I - \gamma P, \qquad \mathcal{L}_1 = b^T P, \qquad \mathcal{M}_0 = \gamma I, \qquad \mathcal{M}_1 = 0, \quad \gamma = \left(\max_i p_{ii} \right)^{-1}$$

where $P = (p_{ij}) = A^{-1}$ [22, 33].

For a given Runge–Kutta method, finding the optimal value of $c(\alpha, \beta)$ using the Shu-Osher formulation requires solving a nonlinear optimization problem. The theory of absolute monotonicity, on the other hand, provides a purely algebraic characterization of the SSP coefficient, making calculation of $c(\mathbf{A}, \boldsymbol{b})$ trivial. This leads to simplification of the problem of finding optimal methods [21, 23]. For instance, recent investigations of optimal implicit [47] and explicit [45] Runge–Kutta methods with many stages would not have been possible without this simplification. In Sect. 5, we will see that the new theory also leads to a simplified analysis of the SSP properties of spectral deferred correction methods.

For more details regarding the relationship between contractivity, absolute monotonicity, and the Shu-Osher form, see [20, 22, 32, 33]. Higueras has extended this theory to include the case where some elements of A or b may be negative, by considering perturbed Runge–Kutta methods [33]. This is equivalent, in the Shu-Osher representation, to considering $\tilde{F}$.

Spijker has extended the theory of absolutely monotonic methods to the much larger class of general linear methods [81]. These methods combine the approaches of both the multistep and Runge–Kutta methods, by taking function evaluations at multiple steps and multiple stages. These methods can be written in the form

$$y^i = \sum_{j=1}^{l} s_{ij} u_j^{(n-1)} + \Delta t \sum_{j=1}^{m} t_{ij} F(y^j) \quad (1 \le i \le m), \tag{1.33a}$$

$$u_i^{(n)} = y_{m-l+i} \quad (1 \le i \le l), \tag{1.33b}$$

so that the method is determined by the coefficient matrices $\mathbf{S}, \mathbf{T}$. The largest timestep under which an irreducible method of this form preserves strong stability is shown to be $c(\mathbf{S}, \mathbf{T})\Delta t_{\mathrm{FE}}$, where $c(\mathbf{S}, \mathbf{T})$ is the radius of absolute monotonicity of the method, defined as the largest $r \geq 0$ such that

$$(I + r\mathbf{T})^{-1} \text{ exists} \quad \text{and} \tag{1.34a}$$

$$(I + r\mathbf{T})^{-1}\mathbf{S} \geq 0, \tag{1.34b}$$

$$(I + r\mathbf{T})^{-1}\mathbf{T} \geq 0. \tag{1.34c}$$

As usual, inequalities are understood component-wise, and we take $c(\mathbf{S}, \mathbf{T}) = 0$ if $\mathbf{S} \geq 0$ or $\mathbf{T} \geq 0$ is violated. For further details, see [81].

2 Bounds on the SSP Coefficient

For a given ODE it may be possible to satisfy a desired stability requirement using a method that is not SSP or a timestep that violates the SSP timestep restriction. However, the SSP property guarantees that *any* nonlinear stability property will be preserved for *any* ODE provided only that it is satisfied using forward Euler. Clearly, the SSP property is a very strong requirement, and imposes severe restrictions on other properties of a time discretization method. Known results in contractivity theory lead to restrictions on the obtainable SSP coefficient, and order barriers on SSP methods with positive SSP coefficient.

2.1 Runge–Kutta Methods

Many useful results for SSP Runge–Kutta methods were collected in [47]; we briefly review them here. First, we introduce some notation. Let $C^{\mathrm{IRK}}_{m,p}$ ($C^{\mathrm{ERK}}_{m,p}$) denote the optimal radius of absolute monotonicity over the class of implicit (explicit) Runge–Kutta methods with at most m stages and at least order p. Let $R_{m,p}$ denote the optimal radius of absolute monotonicity over all polynomials of degree at most m that approximate the exponential to order at least p (i.e., stability functions of m-stage, order p explicit Runge–Kutta methods). Since absolute monotonicity of a method implies absolute monotonicity of the stability function, then

$$C^{\mathrm{ERK}}_{m,p} \leq R_{m,p} \tag{2.35}$$

for any m, p. This bound is useful because values and properties of $R_{m,p}$ are generally easier to compute. Furthermore, equality in (2.35) is attained in many cases, leading to verification of the optimality of methods found by numerical search.

By considering conditions (1.32) with $r = 0$, we see that $R(\mathbf{A}, \boldsymbol{b}) > 0$ implies $\mathbf{A} \geq 0$ and $\boldsymbol{b} > 0$ [51]. This provides a useful lower bound on the coefficients when searching for optimal SSP Runge–Kutta methods.

The requirement of non-negativity of $\mathbf{A}$ and strict positivity of $\boldsymbol{b}$ leads to restrictions on the stage order of an SSP method. The stage order $\tilde{p}$ is a lower bound on the order of convergence when a method is applied to arbitrarily stiff problems. Low stage order may lead to *order reduction*, i.e. slow convergence, when computing solutions of stiff ODEs. Any Runge–Kutta method with non-negative coefficients $\mathbf{A} \geq 0$ must have stage order $\tilde{p} \leq 2$. Furthermore, if it has $\tilde{p} = 2$, then $\mathbf{A}$ must have a zero row [51]. There is also a relationship between the classical order and the stage order: a Runge–Kutta method with weights $\boldsymbol{b} > 0$

must have stage order $\tilde{p} \geq \lfloor \frac{p-1}{2} \rfloor$ [51]. These results apply to all Runge–Kutta methods; however, when dealing with explicit methods, stage order is limited to $\tilde{p} \leq 1$ whether or not one requires non-negative coefficients [14, 51]. Combining the foregoing results, it follows that any irreducible Runge–Kutta method with $c(\mathbf{A}, \boldsymbol{b}) > 0$ can be no greater than fourth order accurate if it is explicit and no greater than sixth order accurate if it is implicit [51].

Hence implicit methods allow for higher stage order, and higher classical order than explicit methods. However, implicit Runge–Kutta methods that are unconditionally SSP must have order at most one i.e. $C^{\text{IRK}}_{m,p} < \infty$ for $p > 1$ [29, 80]. This result is in contrast with linear stability and B-stability, where some high-order implicit methods (i.e., the A-stable methods and the algebraically stable methods, respectively) are unconditionally stability preserving.

2.1.1 Singly Implicit and Diagonally Implicit Methods

An m-stage Runge–Kutta method applied to a system of N ODEs typically requires the solution of a system of mN equations. When the system results from the semi-discretization of a system of nonlinear PDEs, N is typically very large and the equations are nonlinear, making their solution very expensive. Using a transformation involving the Jordan form of $\mathbf{A}$, the amount of work can be reduced [4]. This is especially efficient for *singly implicit* (SIRK) methods (those methods for which $\mathbf{A}$ has only one distinct eigenvalue), because the necessary matrix factorizations can be reused. On the other hand, *diagonally implicit* (DIRK) methods, for which $\mathbf{A}$ is lower triangular, can be implemented efficiently without transforming to the Jordan form of $\mathbf{A}$. The class of *singly diagonally implicit* (SDIRK) methods, which are both singly implicit and diagonally implicit (i.e., $\mathbf{A}$ is lower triangular with all diagonal entries identical), incorporates both of these advantages. For details on efficient implementation of implicit Runge–Kutta methods see, e.g., [15].

Contractivity theory provides barriers for the special case of singly implicit and diagonally implicit methods. These barriers were reviewed in [47]. SSP SDIRK methods have the same order barrier ($p \leq 4$) as explicit methods. For DIRK methods in general, and for SIRK methods that are SSP, the order of an m-stage method is at most $m + 1$. Furthermore, for m-stage SIRK methods of order $p \geq 5$, we find that $c(\mathbf{A}, \boldsymbol{b})$ is bounded by the optimal linear SSP coefficient of m-stage explicit Runge–Kutta methods of the same order (see [45, 50] for values of these optimal coefficients).

2.2 Linear Multistep Methods

Lenferink conducted an extensive study of contractive linear multistep methods [56, 57]. As discussed above, the optimal contractive methods are also optimal SSP methods. Furthermore, for explicit methods of this type, the SSP coefficient is simply the threshold factor. For implicit methods, the two factors coincide in many cases, and the difference between them is generally quite small [57]. For explicit s-step methods of order p, it holds that $c \leq \frac{s-p}{s-1}$ for $s > 1$; for implicit methods of order $p > 1$, $c \leq 2$. While there appears to be no limit to the order of accuracy of SSP linear multistep methods, high order accurate methods of this type are subject to very small timestep restrictions or very many steps (see Tables 6 and 8) and require very many steps.

2.3 General Linear Methods

Bounds on the SSP coefficient for general linear methods have not been previously given. Here we present a recent result from [46] on the upper bound of the SSP coefficient of

explicit general linear methods. Consider a general linear method involving at most m stages and s steps. When applied to a linear constant coefficient problem, the method takes the form

$$u^{n+1} = \psi_0(\Delta t L)u^n + \psi_1(\Delta t L)u^{n-1} + \cdots + \psi_{s-1}(\Delta t L)u^{n-s+1},$$

where each ψ_i is a polynomial of degree at most m:

$$\psi_i = \sum_{j=0}^{m} a_{ij} z^j, \quad 1 \le i \le s. \tag{2.36}$$

The theory reviewed in the previous section tells us that the strong stability preserving coefficient c is at most equal to the threshold factor R. We will show that if the method is at least first order accurate, then the threshold factor is at most equal to the number of stages m.

The method is first order accurate if the following conditions are satisfied:

$$\sum_{i=1}^{s} a_{i0} = 1, \tag{2.37a}$$

$$\sum_{i=1}^{s} (a_{i1} + (s-i)a_{i0}) = s. \tag{2.37b}$$

Let R denote the threshold factor of the method. Since each ψ_i is absolutely monotonic on the interval $(-R, 0]$, we can write

$$\psi_i = \sum_{j=0}^{m} \gamma_{ij} \left(1 + \frac{z}{R}\right)^j \quad \text{with } \gamma_{ij} \ge 0. \tag{2.38}$$

Equating the right hand sides of (2.36) and (2.38) gives the following relation between the coefficients a_{ij} and γ_{ij}:

$$a_{il} = \frac{1}{l! R^l} \sum_{j=0}^{m} \gamma_{ij} \prod_{n=0}^{l-1} (j-n). \tag{2.39}$$

Substituting (2.39) in (2.37) yields

$$\sum_{i=1}^{s} \sum_{j=0}^{m} \gamma_{ij} = 1, \tag{2.40a}$$

$$\sum_{i=1}^{s} \sum_{j=0}^{m} \gamma_{ij} (j + R(s-i)) = sR. \tag{2.40b}$$

Subtracting sm times (2.40a) from (2.40b) gives

$$\sum_{i=1}^{s} \sum_{j=0}^{m} \gamma_{ij} \, (j + R(s-i) - sm) = s(R-m). \tag{2.41}$$

Since, for $1 \le i \le s$, $0 \le j \le m$,

$$j + R(s-i) - sm = (j-m) + R(1-i) + (R-m)(s-1) \le (R-m)(s-1), \quad (2.42)$$

we have

$$\begin{aligned} s(R-m) &= \sum_{i=1}^{s}\sum_{j=0}^{m} \gamma_{ij}\,(j + R(s-i) - sm) \\ &\le (s-1)(R-m)\sum_{i=1}^{s}\sum_{j=0}^{m}\gamma_{ij} \\ &= (s-1)(R-m), \end{aligned}$$

which implies that $R \le m$.

Since the SSP coefficient $c(\mathbf{S}, \mathbf{T})$ (relevant to nonlinear problems) is no larger than the threshold factor R, this implies that the SSP coefficient of any explicit general linear method is at most equal to its number of stages.

Note that this result can also be shown using Theorem 3.1 in [41], which deals with the radius of stability of a method.

3 Optimal SSP Runge–Kutta Methods

In this section we review the best available explicit and implicit SSP Runge–Kutta methods. In addition to optimization of the effective SSP coefficient, we pay attention (for explicit methods) to those that have low-storage implementations—an important consideration when solving PDEs. For implicit methods, we will pay particular attention to methods which can be implemented efficiently (i.e., diagonally implicit and singly implicit methods). In both cases, we will see that the optimal SSP methods often possess these favorable secondary properties.

Again we focus on methods with positive coefficients, because they are much more widely used. For SSP Runge–Kutta methods with negative coefficients, see [21, 25, 27, 29, 34, 72, 75, 77, 79, 81].

3.1 Optimal Explicit SSP Runge–Kutta Methods

In this section, we present the best known explicit SSP Runge–Kutta methods. From the results above (Sect. 2.1) we know that these methods have order at most four. We will also discuss low-storage implementations of these methods. In the following, SSPRK (m, p) denotes the optimal m-stage, p-th Runge–Kutta order method.

The two-stage second-order and three-stage third-order optimal explicit SSPRK schemes were presented in [79]. These are optimal among all Runge–Kutta methods with their respective order and number of stages [28]. They are:
SSPRK (2,2):

$$\begin{aligned} u^{(1)} &= u^n + \Delta t F(u^n), \\ u^{n+1} &= \frac{1}{2}u^n + \frac{1}{2}u^{(1)} + \frac{1}{2}\Delta t F(u^{(1)}), \end{aligned} \quad (3.1)$$

and SSPRK (3,3):

$$\begin{aligned} u^{(1)} &= u^n + \Delta t F(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t F(u^{(1)}), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t F(u^{(2)}). \end{aligned} \tag{3.2}$$

Although both these methods have SSP coefficient $c = 1$, which permits a timestep of the same size as forward Euler would permit, it is clear that the computational cost is double and triple (respectively) that of the forward Euler. Thus, we find it useful to define the *effective SSP coefficient* as $c_{\text{eff}} = \frac{c}{m}$, where m is the number of stages. In the case of SSPRK (2,2) and SSPRK (3,3) the effective SSP coefficient is $c_{\text{eff}} = \frac{1}{2}$ and $c_{\text{eff}} = \frac{1}{3}$, respectively. For a given order of accuracy, optimal SSPRK methods with more stages typically have larger effective SSP coefficient. For instance, the optimal five-stage, third order method [82]

$$\begin{aligned} u^{(1)} &= u^n + 0.37726891511710\Delta t F(u^n), \\ u^{(2)} &= u^{(1)} + 0.37726891511710\Delta t F(u^{(1)}), \\ u^{(3)} &= 0.56656131914033u^n + 0.43343868085967u^{(2)} + 0.16352294089771\Delta t F(u^{(2)}), \\ u^{(4)} &= 0.09299483444413u^n + 0.00002090369620u^{(1)} + 0.90698426185967u^{(3)} \\ &\quad + 0.00071997378654\Delta t F(u^n) + 0.34217696850008\Delta t F(u^{(3)}), \\ u^{(5)} &= 0.00736132260920u^n + 0.20127980325145u^{(1)} + 0.00182955389682u^{(2)} \\ &\quad + 0.78952932024253u^{(4)} + 0.00277719819460\Delta t F(u^n) \\ &\quad + 0.00001567934613\Delta t F(u^{(1)}) + 0.29786487010104\Delta t F(u^{(1)}) \end{aligned} \tag{3.3}$$

has $c_{\text{eff}} = 0.53$, larger than that of SSPRK (3,3). This method was guaranteed optimal in [72].

There exists no explicit four-stage fourth-order Runge Kutta method with $c > 0$ [28, 51]; The numerically optimal five stage method was found in [51] and again independently in [82], and guaranteed optimal [72].

SSPRK (5,4):

$$\begin{aligned} u^{(1)} &= u^n + 0.391752226571890\Delta t F(u^n), \\ u^{(2)} &= 0.444370493651235u^n + 0.555629506348765u^{(1)} \\ &\quad + 0.368410593050371\Delta t F(u^{(1)}), \\ u^{(3)} &= 0.620101851488403u^n + 0.379898148511597u^{(2)} \\ &\quad + 0.251891774271694\Delta t F(u^{(2)}), \\ u^{(4)} &= 0.178079954393132u^n + 0.821920045606868u^{(3)} \\ &\quad + 0.544974750228521\Delta t F(u^{(3)}), \\ u^{n+1} &= 0.517231671970585u^{(2)} \end{aligned}$$

$$+ 0.096059710526147u^{(3)} + 0.063692468666290\Delta t F(u^{(3)})$$
$$+ 0.386708617503269u^{(4)} + 0.226007483236906\Delta t F(u^{(4)}).$$

This method has SSP coefficient $c = 1.508$, and effective SSP coefficient $c_{\text{eff}} = 0.302$ which means that this method is higher order and only slightly less efficient than the popular SSPRK (3,3).

3.1.1 Low-storage Considerations

Storage is an important consideration for large scale scientific computing in three space dimensions. A naive implementation of an m-stage Runge–Kutta method requires $m + 1$ memory registers. However, if certain algebraic relations between the coefficients are satisfied, the method may be implemented with fewer registers. Three such types of relations have been exploited in the literature [44, 45, 94]. The resulting types of low-storage methods make different important assumptions on the manner in which F is evaluated.

Consider two storage registers, q1 and q2, each of size N, where N denotes the size of the ODE system. The low-storage methods of Williamson [94] assume that it is possible to make assignments of the form

$$\texttt{q1} := \texttt{q1} + \texttt{F(q2)},$$

without using (much) additional storage beyond the two registers. As noted in [44], this requires that the evaluation be done in 'piecemeal fashion'. This is natural, for instance, if F corresponds to a spatial discretization of a PDE where the spatial stencil is localized, which is usually the case for semi-discretizations of hyperbolic PDEs. We refer to these as MN methods, where M represents the number of storage registers required.

The low-storage methods of van der Houwen type [44] make instead the assumption that it is possible to make assignments of the form

$$\texttt{q1} := \texttt{F(q1)},$$

without much additional storage beyond a single register. This is apparently reasonable for compressible Navier-Stokes calculations [44]. Following the terminology of [44], we refer to these as MR methods, where M represents the number of storage registers required.

Some SSP low storage implementations of these two types were studied in [28, 29, 72, 75]. In [72], Ruuth presented ten low storage methods, of order $p = 3$ and $p = 4$ and $m = 3, 4, 5$ stages resulting from numerical optimization. Some of these methods are guaranteed optimal, others are the best found in extensive numerical searches.

In [45] some low-storage implementations were presented that require the assumption that it is possible to take a forward Euler step

$$\texttt{q1} := \texttt{q1} + \texttt{F(q1)},$$

without much additional storage beyond a single register. This assumption is quite strong, but valid for spatial discretizations with localized stencils, with careful programming. We refer to these as MS methods.

In the following, a method requiring M storage registers is referred to as an MN (MR, MS) method. Sometimes it is necessary to retain the value of the solution at the previous timestep, usually in order to restart the step if some accuracy or stability requirement is violated. While most low-storage methods will require an additional register in this case, some will not. Such methods are denoted by MN* (or MR*, MS*).

Second Order Methods SSPRK $(m, 2)$

$$\beta_{i,i-1} = \begin{cases} \frac{1}{m-1} & 1 \le i \le m-1, \\ \frac{1}{m} & i = m, \end{cases} \tag{3.4a}$$

$$\alpha_{i,i-1} = \begin{cases} 1 & 1 \le i \le m-1, \\ \frac{m-1}{m} & i = m, \end{cases} \tag{3.4b}$$

$$\alpha_{m,0} = \frac{1}{m}. \tag{3.4c}$$

The first method in this family ($m = 2$) was proposed in [79]. The full family was found in [51] and again independently in [26, 82]. These methods are SSP with $c = m - 1$, so they have $c_{\text{eff}} = \frac{m-1}{m}$. They can be implemented using only three memory registers, even if the previous timestep must be retained. These methods can be implemented in 3N* or 3R* form. In [45], a 2S* implementation was given.

Third Order Methods Optimal three- and four-stage third-order SSP Runge–Kutta methods, originally reported in [79] and [51], respectively, can be implemented in 2S* (or 3N*/3R*) fashion. The four-stage method is 50% more efficient and requires the same amount of memory as the three-stage method.

Further savings in storage costs can be obtained if the solution at the previous timestep can be discarded. These methods are denoted without an asterisk, e.g. 2N or 3N methods. Of course, the previous timestep can be retained at the cost of using one more register. Third-order methods of this type were studied in [72]; the best 2R method has $c_{\text{eff}} = 0.297$; the best 3R method has $c_{\text{eff}} = 0.513$. These are more efficient than the optimal 2N and 3N methods, respectively. In [45], a family of third-order 2S SSP Runge–Kutta methods with $m = n^2$ stages (for $n > 1$) was discovered:

$$\alpha_{i,i-1} = \begin{cases} \frac{n-1}{2n-1} & i = \frac{n(n+1)}{2}, \\ 1 & \text{otherwise}, \end{cases} \tag{3.5a}$$

$$\alpha_{\frac{n(n+1)}{2}, \frac{(n-1)(n-2)}{2}} = \frac{n}{2n-1}, \qquad \beta_{i,i-1} = \frac{\alpha_{i,i-1}}{n^2 - n}. \tag{3.5b}$$

These methods are optimal in terms of SSP timestep restriction, with SSP coefficient $c = n^2 - n$. The effective SSP coefficient for this method $c_{\text{eff}} = 1 - \frac{1}{n} = 1 - \frac{1}{\sqrt{m}}$, can be made as close to one as desired by taking more stages, without raising the storage cost. Furthermore, the coefficients are simple rational numbers. Note that the four-stage 2S* method mentioned above is the first member of this family. A low-storage implementation of (3.5) is given in [45].

Fourth Order Methods The optimal five-stage, fourth order method was given above. By further increasing the number of stages allowed, we can obtain fourth-order SSP methods with larger timestep restriction. Spiteri and Ruuth [82, 83] and Macdonald [61] developed fourth order methods with up to eight stages. The most efficient (eight-stage) method has $c_{\text{eff}} = 0.518$ and can be implemented in 3N fashion.

In [45] a ten-stage fourth order 2S Runge–Kutta method was found with an effective SSP coefficient greater than any previously known fourth order full-storage method. Additionally, this is the only fourth order SSP method to be analytically proved optimal, because it achieves the optimal bound on ten-stage, fourth order SSP methods for linear problems:

Table 3 Effective SSP coefficients c_{eff} of best known explicit SSP methods, and scaled threshold factors ρ of optimal methods for linear systems. A dash indicates that SSP methods of this type cannot exist. Bold entries indicate methods that have $c_{\text{eff}} = \rho_{m,p}$

m	p					
	$c_{\text{eff}} = c/m$			$\rho_{m,p} = R_{m,p}/m$		
	2	3	4	2	3	4
1	–	–	–	–	–	–
2	**0.5**	–	–	0.5	–	–
3	**0.67**	**0.33**	–	0.67	0.33	–
4	**0.75**	**0.5**	–	0.75	0.5	0.25
5	**0.8**	**0.53**	0.30	0.8	0.53	0.40
6	**0.83**	**0.59**	0.38	0.83	0.59	0.44
7	**0.86**	**0.61**	0.47	0.86	0.61	0.50
8	**0.88**	**0.64**	0.52	0.88	0.64	0.54
9	**0.89**	**0.67**	0.54	0.89	0.67	0.57
10	**0.9**	**0.68**	**0.60**	0.9	0.68	0.60
11	**0.91**	**0.69**	0.59	0.91	0.69	0.62

$c = 6$. Furthermore, the method has simple rational coefficients. The nonzero coefficients are

$$\beta_{i,i-1} = \begin{cases} \frac{1}{6} & i \in \{1..4, 6..9\}, \\ \frac{1}{15} & i = 5, \\ \frac{1}{10} & i = 10, \end{cases} \qquad \alpha_{i,i-1} = \begin{cases} 1 & i \in \{1..4, 6..9\}, \\ \frac{2}{5} & i = 5, \\ \frac{3}{5} & i = 10, \end{cases} \tag{3.6a}$$

$$\beta_{10,4} = \frac{3}{50}, \qquad \alpha_{10,4} = \frac{9}{25}, \tag{3.6b}$$

$$\alpha_{5,0} = \frac{3}{5}, \qquad \alpha_{10,0} = \frac{1}{25}. \tag{3.6c}$$

The abscissas are

$$c = \frac{1}{6} \cdot (0, 1, 2, 3, 4, 2, 3, 4, 5, 6)^T.$$

A 2S implementation of the ten-stage method is:

```
   q1 = u; q2=u;
for i=1:5
   q1 = q1 + dt*F(q1)/6;
end
q2 = 1/25*q2 + 9/25*q1;
q1 = 15*q2-5*q1;
for i=6:9
   q1 = q1 + dt*F(q1)/6;
end
q1 = q2 + 3/5*q1 + 1/10*dt*F(q1);
u=q1;
```

In [45] fourth-order 3S methods with more than ten stages were found that are more efficient than the 2S ten-stage method above; the most efficient has 26 stages and $c_{\text{eff}} \approx 0.696$. Table 4 contains a comparison of the explicit methods described in this section.

Table 4 Properties of some optimal SSP Runge–Kutta methods. An asterisk indicates that the previous timestep can be retained without increasing the required number of registers

Popular method	c_{eff}	Storage	Improved method	c_{eff}	Storage
SSPRK (2,2)	0.500	2N*	SSPRK (m,2)	$1-1/m$	2N*
SSPRK (3,3)	0.333	2N*	SSPRK (n^2,3)	$1-1/n$	2N
SSPRK (5,4)	0.377	3N	SSPRK (10,4)	0.600	2N

For Runge–Kutta methods with many stages, it is important to consider amplification of roundoff errors occurring in the intermediate stages. The associated stability property is referred to as *internal stability*. In [45], it was shown that all the methods in this section are internally stable.

3.1.2 Optimal SSP Runge–Kutta Methods for Linear Constant Coefficient Problems

While SSP methods were developed for nonlinear hyperbolic PDEs, this property has proven useful for linear problems as well. In [58], the authors used the energy method to analyze the stability of Runge–Kutta methods for ODEs resulting from coercive approximations such as those in [24]. Using this method it can be proved, for example, that the fourth order Runge–Kutta method preserved a certain stability property with a CFL number of $\frac{1}{31}$. However, using SSP theory, one easily shows that the same stability property is preserved in the linear case under a CFL number as large as 1. Linear SSP Runge–Kutta methods are thus useful from the point of view of stability analysis. These methods are also of interest in their own right, for solving linear wave equations, such as Maxwell's equations and linear elasticity.

As discussed in Sect. 1.4.1, the conditions for a method to preserve strong stability for linear autonomous systems only are less restrictive than those required for the nonlinear SSP property.

For such problems, the contractivity condition (1.23) and the monotonicity condition (1.6) are clearly equivalent. Recall that the timestep restriction is given in this case by (1.29) with $R = R(\phi)$ [80]. For explicit Runge–Kutta methods with m stages and order p, $\phi(z)$ is a polynomial of degree m that approximates the exponential function to order p near $z = 0$. The problem of finding such optimal methods was first considered in [50], where optimal methods were given for many values of m and p, including $1 \le p \le m \le 10$, and $p \in \{1, 2, 3, 4, m-1, m-2, m-3, m-4\}$ for any m, as well as an algorithm for the computation of the optimal coefficient and method for arbitrary m, p. Unfortunately, the computational cost of this algorithm grows exponentially in m and p. The results for the cases $p \in \{1, 2, m-1, m\}$, which are listed below, were found again independently using a related approach in [26, 29]. Recently, an efficient algorithm for the determination of optimal methods for any m, p was given in [45].

Any m-stage, p-th order SSP Runge–Kutta method (1.15) with nonnegative coefficients $\alpha_{i,k}$ and $\beta_{i,k}$ must have SSP coefficient $c \le m - p + 1$. This barrier is not generally sharp, but the SSP Runge–Kutta methods for linear constant coefficient problems which do attain this barrier are listed below.

SSPRK Linear (m, m): [29] The class of m stage schemes given by:

$$u^{(i)} = u^{(i-1)} + \Delta t L u^{(i-1)}, \quad i = 1, \ldots, m-1,$$
$$u^{(m)} = \sum_{k=0}^{m-2} \alpha_{m,k} u^{(k)} + \alpha_{m,m-1} \left(u^{(m-1)} + \Delta t L u^{(m-1)} \right),$$

where $\alpha_{1,0} = 1$ and

$$\alpha_{m,k} = \frac{1}{k} \alpha_{m-1,k-1}, \quad k = 1, \ldots, m-2,$$
$$\alpha_{m,m-1} = \frac{1}{m!}, \qquad \alpha_{m,0} = 1 - \sum_{k=1}^{m-1} \alpha_{m,k}$$

is an m-order linear Runge–Kutta method which is SSP with threshold factor $R = 1$, which is optimal among all m stage, $p = m$ order SSPRK methods with nonnegative coefficients. The scaled threshold factor is $\rho = \frac{1}{m}$.

SSPRK Linear $(m, 1)$: The m stage, first order SSP Runge–Kutta method given by

$$u^{(0)} = u^n,$$
$$u^{(i)} = \left(1 + \frac{\Delta t}{m} L \right) u^{(i-1)}, \quad i = 1, \ldots, m,$$
$$u^{n+1} = u^{(m)},$$

has threshold factor $R = m$, which is optimal in the class of m stage, order $p = 1$ methods with nonnegative coefficients. This allows for a larger timestep but the computational cost increases correspondingly. This is reflected by the fact that the effective threshold factor is $\rho = 1$, which is equivalent to the forward Euler method.

SSPRK Linear $(m, 2)$: The m stage, second order SSP methods given in (3.4) above have an optimal threshold factor $R = m - 1$. Although these methods were designed for linear problems, they are also nonlinearly second order [82]. Each such method uses m stages to attain the order usually obtained by a 2-stage method, but has optimal threshold factor $R = m - 1$, thus the scaled threshold factor here is $\rho = \frac{m-1}{m}$.

SSPRK Linear $(m, m-1)$: The m stage, order $p = m - 1$ method

$$u^{(0)} = u^n,$$
$$u^{(i)} = u^{(i-1)} + \frac{1}{2} \Delta t L u^{(i-1)}, \quad i = 1, \ldots, m-1,$$
$$u^{(m)} = \sum_{k=0}^{m-2} \alpha_{m,k} u^{(k)} + \alpha_{m,m-1} \left(u^{(m-1)} + \frac{1}{2} \Delta t L u^{(m-1)} \right),$$
$$u^{n+1} = u^{(m)}$$

with the coefficients given by

$$\alpha_{2,0} = 0, \qquad \alpha_{2,1} = 1,$$

$$\alpha_{m,k} = \frac{2}{k}\alpha_{m-1,k-1}, \quad k = 1, \ldots, m-2,$$

$$\alpha_{m,m-1} = \frac{2}{m}\alpha_{m-1,m-2}, \quad \alpha_{m,0} = 1 - \sum_{k=1}^{m-1} \alpha_{m,k}$$

is SSP with optimal (for methods with nonnegative coefficients) threshold factor $R = 2$. The scaled threshold factor for these methods is $\rho = \frac{2}{m}$.

An interesting application of these methods is the case of a constant linear operator with a time dependent forcing term [26, 78]. This case could arise from a linear PDE with time dependent boundary conditions, such as Maxwell's equations (see [8]), which can be written as:

$$u_t = Lu + f(t) \tag{3.7}$$

where $u = [u_i]$ is a vector, $L = [L_{i,j}]$ is a constant matrix and $f(t) = [f_i(t)]$ is a vector of functions of t. The functions $f(t)$ can typically be written as or approximated by

$$f_i(t) = \sum_{j=0}^{n} a_j^i q_j(t) = [Aq(t)]_i$$

where $A = [A_{i,j}] = [a_j^i]$ is a constant matrix and $q(t) = [q_j(t)]$ are a set of functions which have the property that $q'(t) = Dq(t)$, where D is a constant matrix. If so, then (3.7) can be converted to a linear constant-coefficient ODE of the form

$$y_t = My(t), \tag{3.8}$$

where

$$y(t) = \begin{pmatrix} q(t) \\ u(t) \end{pmatrix} \quad \text{and} \quad M = \begin{pmatrix} D & 0 \\ A & L \end{pmatrix}.$$

Thus, an equation of the form (3.7) can be approximated (or given exactly) by a linear constant coefficient ODE, and the SSP Runge–Kutta methods for linear constant coefficient operators can be applied to guarantee that any strong stability properties satisfied with forward Euler will be preserved.

3.2 Optimal Implicit SSP Runge–Kutta Methods

For classical stability properties (such as linear stability or B-stability), implicit methods exist that are stable under arbitrarily large timesteps. Similarly, it can be easily shown that any spatial discretization F which is strongly stable in some norm using the forward Euler method with some finite timestep restriction will be *unconditionally* strongly stable, in the same norm, using the implicit Euler method [32, 40]. However, no unconditionally SSP method has order greater than one [80], although implicit methods may have SSP coefficients significantly larger than those of explicit methods with the same order and number of stages. The question is, then, whether the allowable step-size can be large enough to offset the extra computational effort required in the implicit solution of the resulting system at each iteration.

Recall from Sect. 2.1 that implicit SSP Runge–Kutta methods have order at most six; existence of methods of order up to five was established in [51]. Recently, Ferracina and Spijker investigated optimal singly diagonally implicit methods [23]. They showed that such methods have order at most four, and found optimal methods (by numerical optimization)

of up to order four and up to eight stages. They also conjectured the form of optimal second and third order methods with any number of stages. In [47], fully implicit SSP Runge–Kutta methods were investigated via numerical optimization. This search yielded the first sixth order SSP Runge–Kutta methods, demonstrating that Kraaijevanger's order barrier is sharp. Remarkably, searching among the class of fully implicit methods, the optimal methods of second and third order were found to be singly diagonally implicit; in fact, they were the very methods found already in [23]. The optimal methods of fourth through sixth order were found to be diagonally implicit.

Unfortunately, when we look at the *effective SSP coefficient*, which is defined as the SSP coefficient normalized by the number of stages, we notice that all of these methods turn out to have effective SSP coefficient less than or equal to two, making them probably too inefficient for practical use. We reproduce some of the optimal methods and SSP coefficients here, as they may be of theoretical interest.

In the following, we give modified Shu–Osher arrays for the numerically optimal methods. To simplify implementation, we present modified Shu–Osher arrays in which the diagonal elements of λ are zero. This form is a simple rearrangement and involves no loss of generality. In comparing methods with different numbers of stages, one is usually interested in the relative time advancement per computational cost. For diagonally implicit methods, the computational cost per timestep is proportional to the number of stages. We therefore define the effective SSP coefficient of a method as $\frac{c}{m}$; this normalization enables us to compare the cost of integration up to a given time using DIRK schemes of order $p > 1$. However, for non-DIRK methods of various m, it is much less obvious how to compare computational cost.

Second Order Methods: The numerically optimal second-order method with m stages is

$$\lambda = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & 1 & \ddots & \\ & & \ddots & 0 \\ & & & 1 \end{bmatrix}, \qquad \mu = \begin{bmatrix} \frac{1}{2m} & & & \\ \frac{1}{2m} & \frac{1}{2m} & & \\ & \frac{1}{2m} & \ddots & \\ & & \ddots & \frac{1}{2m} \\ & & & \frac{1}{2m} \end{bmatrix}. \tag{3.9}$$

These methods have SSP coefficient $c = 2m$, and effective SSP coefficient $c_{\text{eff}} = 2$. Note the sparse, bidiagonal modified Shu–Osher arrays, which make these methods efficient to implement. These methods were proved optimal analytically for $m = 1, 2$ in [23], and using BARON [76] for $m = 3$ in [47]. The one-stage method of this class is the implicit midpoint rule, while the m-stage method is equivalent to m successive applications of the implicit midpoint rule [21]. Thus these methods inherit the desirable properties of the implicit midpoint rule such as algebraic stability and A-stability [30]. If these methods are indeed optimal, this would imply that the effective SSP coefficient of any Runge–Kutta method of order greater than one is at most equal to two.

Third-Order Methods: For $m \geq 2$ the numerically optimal methods have coefficients

$$\mu = \begin{bmatrix} \mu_{11} & & & \\ \mu_{21} & \ddots & & \\ & \ddots & \mu_{11} & \\ & & \mu_{21} & \mu_{11} \\ & & & \mu_{m+1,m} \end{bmatrix}, \qquad \lambda = \begin{bmatrix} 0 & & & \\ 1 & \ddots & & \\ & \ddots & 0 & \\ & & 1 & 0 \\ & & & \lambda_{m+1,m} \end{bmatrix}, \tag{3.10a}$$

Table 5 Effective SSP coefficients of best known implicit Runge–Kutta methods. A dash indicates that SSP methods of this type cannot exist. A blank space indicates that no SSP methods of this type were found

Order	Number of stages				
	2	3	4	5	6
1	2	–	–	–	–
2	2	1.37	–	–	–
3	2	1.61	0.68	–	–
4	2	1.72	1.11	0.29	–
5	2	1.78	1.21	0.64	–
6	2	1.82	1.30	0.83	0.030
7	2	1.85	1.31	0.89	0.038
8	2	1.87	1.33	0.94	0.28
9	2	1.89	1.34	0.99	0.63
10	2	1.90	1.36	1.01	0.81
11	2	1.91	1.38	1.03	0.80

where

$$\mu_{11} = \frac{1}{2}\left(1 - \sqrt{\frac{m-1}{m+1}}\right), \qquad \mu_{21} = \frac{1}{2}\left(\sqrt{\frac{m+1}{m-1}} - 1\right), \tag{3.10b}$$

$$\mu_{m+1,m} = \frac{m+1}{m(m+1+\sqrt{m^2-1})}, \qquad \lambda_{m+1,m} = \frac{(m+1)(m-1+\sqrt{m^2-1})}{m(m+1+\sqrt{m^2-1})}. \tag{3.10c}$$

These methods have SSP coefficient $c = m - 1 + \sqrt{m^2 - 1}$. The $m = 2$ method was shown to be optimal in [23].

Fourth- Through Sixth- Order Methods All numerically optimal methods of order $4 \leq p \leq 6$ we found in [47] are diagonally implicit. We list effective SSP coefficients of the numerically optimal methods in Table 5. We refer to Table 3, which contains the effective coefficients of optimal explicit methods, for comparison. Many of these implicit methods have representations that allow for very efficient implementation in terms of storage. The coefficients of these representations are available online [48].

The SSP condition provides a guarantee of other necessary properties. When considering implicit Runge–Kutta methods, it is important to determine whether there exists a unique solution of the stage equations. The strong stability preserving timestep restriction turns out to be sufficient for this as well [51, Theorem 7.1]. Furthermore, the SSP condition serves to guarantee that the errors introduced in the solution of the stage equations due to numerical roundoff and (for implicit methods) errors in the implicit solve are not unduly amplified [51, Theorem 7.2].

4 Multistep Methods

In this section, we consider the SSP properties of linear multistep methods. We consider only methods with nonnegative coefficients. For results on linear multistep methods with negative coefficients see [25, 29, 40, 73, 79]. The analysis of the SSP property for multistep methods is much simpler than for Runge–Kutta methods, largely because the form (1.16) for these methods is unique. However, the resulting timestep restrictions are very small and

Table 6 SSP coefficients of some optimal explicit SSP linear multistep methods

	1	2	3	4	5	6	7
1	1.000						
2	1.000						
3	1.000	0.500					
4	1.000	0.667	0.333				
5	1.000	0.750	0.500	0.021			
6	1.000	0.800	0.583	0.165			
7	1.000	0.833	0.583	0.282	0.038		
8	1.000	0.857	0.583	0.359	0.145		
9	1.000	0.875	0.583	0.393	0.228		
10	1.000	0.889	0.583	0.421	0.282	0.052	
11	1.000	0.900	0.583	0.443	0.317	0.115	
12	1.000	0.909	0.583	0.460	0.345	0.175	0.018
13	1.000	0.917	0.583	0.474	0.370	0.210	0.077
14	1.000	0.923	0.583	0.484	0.390	0.236	0.116
15	1.000	0.929	0.583	0.493	0.406	0.259	0.154
16	1.000	0.933	0.583	0.501	0.411	0.276	0.177
17	1.000	0.938	0.583	0.507	0.411	0.291	0.198
18	1.000	0.941	0.583	0.513	0.411	0.304	0.217
19	1.000	0.944	0.583	0.517	0.411	0.314	0.232
20	1.000	0.947	0.583	0.521	0.411	0.322	0.246

exclude many commonly used methods, which has led to the consideration of methods with particular starting procedures.

4.1 Explicit Linear Multistep Methods

As discussed in Sect. 1.4.2, for explicit linear multistep methods the requirement of absolute monotonicity leads immediately to the same conditions as the approach of writing the method in terms of convex combinations of forward Euler steps. In other words, the criteria for the method to be SSP turn out to be the same whether one considers linear or nonlinear problems.

It was shown in [29] that for $s \geq 2$, there is no s-step, s-th order SSP method with all non-negative β_i, and there is no s step SSP method of order $(s+1)$. Thus, we must consider increasing the number of steps to improve the timestep restriction for SSP multistep methods. In this case, adding steps may increase the SSP coefficient but does not require additional computation, only additional storage.

Optimal contractive explicit linear multistep methods were investigated by Lenferink [56], who discovered many interesting properties of these methods and computed optimal methods for up to 20 stages and 7th order accuracy. The results are reproduced here in Table 6. The optimal s-step second-order method was shown to have coefficients

$$\alpha_1 = \frac{(s-1)^2 - 1}{(s-1)^2}, \qquad \alpha_s = \frac{1}{(s-1)^2}, \qquad \beta_1 = \frac{s}{s-1}$$

and SSP coefficient $c = \frac{s-2}{s-1}$ (as usual, the unlisted coefficients are zero).

Table 7 Coefficients of some optimal explicit SSP linear multistep methods

Steps m	Order p	SSP coefficient c	α_i β_i
4	3	$\frac{1}{3}$	$\frac{16}{27}, 0, 0, \frac{11}{27}$ $\frac{16}{9}, 0, 0, \frac{4}{9}$
5	3	$\frac{1}{2}$	$\frac{25}{32}, 0, 0, 0, \frac{7}{32}$ $\frac{25}{16}, 0, 0, 0, \frac{5}{16}$
6	3	0.5828	0.850708871672579, 0, 0, 0, 0.030664864534383, 0.118626263793039 1.459638436015276, 0, 0, 0, 0.052614491749200, 0.203537849338252
5	4	0.0212	0.048963857415019, 0, 0.008344481263315, 0.899467614699687 2.310657177904340, 0, 0.393785059937890, 2.039789323349077, 0
6	4	0.1648	0.342460855717007, 0, 0, 0.191798259434736, 0.093562124939008, 0.372178759909247 2.078553105578060, 0, 0, 1.164112222279710, 0.567871749748709, 0

The coefficients of some optimal third- and fourth-order methods are listed in Table 7. Note that, as with Runge–Kutta methods, we could also consider introducing the downwind operator $\tilde{F}$ to improve the SSP coefficient, but as before this approach typically doubles the number of computations required.

The restrictive SSP coefficients observed in the SSP multistep methods are not surprising, considering that we require the SSP property to hold for *arbitrary* starting values. An illustration of the difficulty is given in [40]: Consider the simple example of the well-known BDF2 method applied to the problem $u'(t) = 0$:

$$u_2 = \frac{4}{3}u_1 - \frac{1}{3}u_0.$$

Clearly, this method is not SSP (α_2 is negative!). In other words, it is not always possible to obtain $\|u_2\| \leq \|u_0\|$ whenever $\|u_1\| \leq \|u_0\|$. However, it is also clear that the only relevant choice for this problem is $u_1 = u_0$, and in this case we do obtain (trivially) $\|u_2\| \leq \|u_0\|$. Using this idea, Hundsdorfer, Ruuth, and Spiteri [40] examined the required step-size for several multistep methods with particular starting procedures. Rather than satisfying a strict monotonicity property, these methods guarantee the boundedness property

$$\|u^n\| \leq M\|u^0\|$$

where M is a constant depending on the starting procedures. Methods of this type were further considered in [38, 73], and methods of up to sixth order were given with reasonably large timestep coefficients. We give here the coefficients of the methods that performed best in numerical tests.

The three-step, third order method has timestep coefficient $C_{LM} = 0.537$ and

$$\alpha_1 = 1.908535476882378, \quad \alpha_2 = -1.334951446162515, \quad \alpha_3 = 0.426415969280137,$$
$$\beta_1 = 1.502575553858997, \quad \beta_2 = -1.654746338401493, \quad \beta_3 = 0.670051276940255.$$

The four-step, fourth order method has timestep coefficient $C_{LM} = 0.458$ and

$$\begin{aligned}
\alpha_1 &= 2.628241000683208, & \beta_1 &= 1.618795874276609,\\
\alpha_2 &= -2.777506277494861, & \beta_2 &= -3.052866947601049,\\
\alpha_3 &= 1.494730011212510, & \beta_3 &= 2.229909318681302,\\
\alpha_4 &= -0.345464734400857, & \beta_4 &= -0.620278703629274.
\end{aligned}$$

Both these methods have fractional coefficient representations which may be more convenient [39]. This creative approach to SSP multistep methods demonstrates that the SSP criteria may sometimes be relaxed or replaced by other conditions on the method.

4.2 Implicit Multistep Methods

For implicit linear multistep methods, the conditions for SSP are stronger than those required for absolute monotonicity of the stability function [57]. Hence it follows from Spijker's work [80] that there are no unconditionally SSP implicit linear multistep methods of order greater than one (see also [29] for a different proof of this). Furthermore, it follows from a result due to Lenferink [57] that any linear multistep method of order $p > 1$ has SSP coefficient no greater than $c = 2$ [40].

It is interesting to note that this bound is actually obtained, for example, by the trapezoidal method. If we wish to compare the efficiency of the trapezoidal method, which involves a system of equations with only one function evaluation and has SSP coefficient $c = 2$, with that of the explicit Runge–Kutta method SSPRK (2,2), which requires two function evaluations and has a SSP coefficient $c = 1$, we notice that the explicit method requires four times as many function evaluations per unit time. However, the cost of solving the implicit system of equations is usually greater than the cost of four explicit function evaluations, so that the explicit method is more computationally efficient.

Lenferink determined the optimal methods of up to twenty steps and order eight [57]; his results are shown in Table 8. Hundsdorfer, Ruuth and Spiteri [40] studied the case of implicit two step methods with different starting procedures, to see if this approach provides a benefit similar to that seen in explicit multistep methods. Even with suitable starting procedures, the step-size restrictions for the implicit multistep methods are hardly better than those of explicit methods. Hundsdorfer and Ruuth [38] showed that, in fact, methods of this type with order greater than one are subject to the same maximal SSP coefficient of two. Thus, implicit SSP multistep methods feature step-size restrictions that are too severe to make the use of these methods efficient.

5 Deferred Correction Methods

In this section we survey the recent work [60] on the study of the SSP property of a newly developed time discretization technique, namely the (spectral) deferred correction (DC) method constructed in [17]. An advantage of this method is that it is a one step method, just like the Runge–Kutta methods, but it can be constructed easily and systematically for any order of accuracy. This is in contrast to Runge–Kutta methods which are more difficult to construct for higher order of accuracy, and to multistep methods which need more storage space and are more difficult to restart with a different choice of the timestep Δt. Linear stability, such as the A-stability, $A(\alpha)$-stability, or L-stability issues for the DC methods were studied in, e.g. [17, 65, 95].

Table 8 SSP coefficients of optimal implicit SSP linear multistep methods

	1	2	3	4	5	6	7	8
1	∞	2.000						
2	∞	2.000	1.000					
3	∞	2.000	1.500	1.000				
4	∞	2.000	1.667	1.243	0.667			
5	∞	2.000	1.750	1.243	0.796	0.500		
6	∞	2.000	1.800	1.243	0.929	0.660	0.300	
7	∞	2.000	1.833	1.243	1.006	0.784	0.468	0.197
8	∞	2.000	1.857	1.243	1.052	0.868	0.550	0.345
9	∞	2.000	1.875	1.243	1.084	0.905	0.642	0.443
10	∞	2.000	1.889	1.243	1.106	0.905	0.690	0.533
11	∞	2.000	1.900	1.243	1.123	0.905	0.733	0.580
12	∞	2.000	1.909	1.243	1.136	0.905	0.764	0.625
13	∞	2.000	1.917	1.243	1.147	0.905	0.781	0.662
14	∞	2.000	1.923	1.243	1.155	0.905	0.795	0.692
15	∞	2.000	1.929	1.243	1.162	0.905	0.806	0.714
16	∞	2.000	1.933	1.243	1.168	0.905	0.815	0.719
17	∞	2.000	1.938	1.243	1.174	0.905	0.823	0.719
18	∞	2.000	1.941	1.243	1.178	0.905	0.829	0.719
19	∞	2.000	1.944	1.243	1.182	0.905	0.835	0.719
20	∞	2.000	1.947	1.243	1.186	0.905	0.839	0.719

The $(s+1)$-th order DC time discretization of (1.2) can be formulated as follows. We first divide the timestep $[t^n, t^{n+1}]$ where $t^{n+1} = t^n + \Delta t$ into s subintervals by choosing the points $t^{(m)}$ for $m = 0, 1, \ldots, s$ such that $t^n = t^{(0)} < t^{(1)} < \cdots < t^{(m)} < \cdots < t^{(s)} = t^{n+1}$. We use $\Delta t^{(m)} = t^{(m+1)} - t^{(m)}$ to denote the sub-timestep and $u_k^{(m)}$ to denote the k-th order approximation to $u(t^{(m)})$. The nodes $t^{(m)}$ can be chosen equally spaced, or as the Chebyshev Gauss-Lobatto nodes on $[t^n, t^{n+1}]$ for high order accurate DC schemes to avoid possible instability associated with interpolation on equally spaced points. Starting from u^n, the DC algorithm to calculate u^{n+1} is in the following.

Compute the initial approximation
$u_1^{(0)} = u^n$.
Use the forward Euler method to compute a first order accurate approximate solution u_1 at the nodes $\{t^{(m)}\}_{m=1}^s$:
For $m = 0, \ldots, s-1$

$$u_1^{(m+1)} = u_1^{(m)} + \Delta t^{(m)} F(u_1^{(m)}). \tag{5.1}$$

Compute successive corrections
For $k = 1, \ldots, s$
$u_{k+1}^{(0)} = u^n$.
For $m = 0, \ldots, s-1$

$$u_{k+1}^{(m+1)} = u_{k+1}^{(m)} + \theta_k \Delta t^{(m)} (F(u_{k+1}^{(m)}) - F(u_k^{(m)})) + I_m^{m+1}(F(u_k)), \tag{5.2}$$

where

$$0 \le \theta_k \le 1 \tag{5.3}$$

and $I_m^{m+1}(L(u_k))$ is the integral of the s-th degree interpolating polynomial on the $s+1$ points $(t^{(\ell)}, L(u_k^{(\ell)}))_{\ell=0}^s$ over the subinterval $[t^{(m)}, t^{(m+1)}]$, which is the numerical quadrature approximation of

$$\int_{t^{(m)}}^{t^{(m+1)}} F(u(\tau))d\tau. \tag{5.4}$$

Finally we have $u^{n+1} = u_{s+1}^{(s)}$.

The scheme described above with $\theta_k = 1$ is the one discussed in [17, 65]. In [95], the scheme is also discussed with general $0 \le \theta_k \le 1$ to enhance linear stability. The term with the coefficient θ_k does not change the order of accuracy.

In [60], the SSP properties of the DC time discretization for the second, third and fourth order accuracy ($s = 1, 2, 3$), were studied. This is just a preliminary study, as the real advantage of the DC time discretization is expected to show more clearly for much higher order of accuracy (the *spectral* DC method). The findings in [60] can be summarized below:

- The second order ($s = 1$) DC time discretization has no subgrid point inside the interval $[t^n, t^{n+1}]$, and it is identical to the optimal second order Runge–Kutta SSP scheme (3.1).
- For the third order ($s = 2$) DC time discretization, there is only one subgrid point inside the interval $[t^n, t^{n+1}]$. By symmetry, this point should be placed in the middle, that is, $t^{(0)} = t^n$, $t^{(1)} = t^n + \frac{1}{2}\Delta t$, $t^{(2)} = t^{n+1}$.

  A numerical optimization procedure can then be performed to search for the SSP scheme with the largest SSP coefficient. Unfortunately, it seems that negative β must appear hence the operator $\tilde{F}$ must be used. A SSP scheme with 10 evaluations of F or $\tilde{F}$ is found to have a SSP coefficient $c = 1.2956$. Several other third order SSP DC schemes are also found in [60] within specific subclasses, however none of them has an impressive SSP coefficient.
- For the fourth order ($s = 3$) DC time discretization, there are two subgrid points inside the interval $[t^n, t^{n+1}]$. By symmetry, these two points should be placed at $t^{(1)} = t^n + a\Delta t$ and $t^{(2)} = t^n + (1-a)\Delta t$ respectively for $0 < a < \frac{1}{2}$. For example, the choice $a = \frac{5-\sqrt{5}}{10}$ would generate the standard Chebyshev Gauss-Lobatto nodes.

  A numerical optimization procedure can then be performed to search for the SSP scheme with the largest SSP coefficient. Unfortunately, it again seems that negative β must appear hence the operator $\tilde{F}$ must be used. A SSP scheme with 17 evaluations of F or $\tilde{F}$ is found to have a SSP coefficient $c = 1.0319$. Several other fourth order SSP DC schemes are also found in [60] within specific subclasses, however none of them has an impressive SSP coefficient.

It would seem from the results in [60] that low order DC schemes are not competitive in terms of SSP properties when comparing with Runge–Kutta methods. It would be interesting to explore higher order DC schemes to see if there is any advantage there. The numerical optimization procedure can be applied to DC schemes of any order to explore their SSP property. The algebra and computational cost for this procedure become very complicated, even for the fourth order methods considered in [60], if the traditional SSP theory is used. However, the analysis is relatively straightforward using the theory of absolutely monotonic methods, whose development we have reviewed in the present work.

Observe that the method (5.1)–(5.2) can be reinterpreted as a Runge–Kutta method. It is easiest to write it in the Shu-Osher form. This involves nothing more than a change of notation, relabeling $u_{k+1}^{(m)}$ as $u^{(sk+m)}$. Comparison of the two forms reveals that the non-zero

coefficients are

$$\begin{aligned}
&\alpha_{i,i-1} = 1, \\
&\beta_{i,i-1} = \Delta t^{(i-1)}/\Delta t, \qquad 1 \le i \le s, \\
&\alpha_{sk+1,0} = 1, \\
&\beta_{sk+1,s(k-1)+i} = C_i^0/\Delta t, \qquad 1 \le k \le s,\ 0 \le i \le s, \\
&\alpha_{sk+m+1,sk+m} = 1, \\
&\beta_{sk+m+1,s(k-1)} = C_i^m/\Delta t, \qquad 1 \le m \le s-1, 1 \le k \le s, 0 \le i \ne m \le s, \\
&\beta_{sk+m+1,sk+m} = \theta_k \Delta t^{(m)}/\Delta t, \\
&\beta_{sk+m+1,s(k-1)+m} = C_m^m/\Delta t - \theta_k \Delta t^{(m)}/\Delta t, \qquad (1 \le k \le s; 1 \le m \le s-1),
\end{aligned}$$

where

$$I_m^{m+1}(L(u_k)) = \sum_{i=0}^{s} C_i^m u_k^{(i)}.$$

Using this, we can immediately obtain the Butcher array $(\mathbf{A}, \boldsymbol{b})$ and consider the much simpler problem of optimizing $c(\mathbf{A}, \boldsymbol{b})$ over the free parameters. For instance, for third order methods, this leads to a two-parameter optimization problem; the same problem was written (after some work) in terms of sixteen free parameters in [60]. For fourth order methods the current approach leads to a six-parameter problem versus 69 parameters in [60].

Noting that spectral DC methods can be written as explicit Runge–Kutta methods, we can immediately conclude that downwind operators will be required in order for explicit spectral DC methods to be SSP if they are of order greater than four. Similarly, implicit spectral DC methods cannot be SSP without downwinding if their order exceeds six.

6 Conclusions

SSP time discretizations were introduced for use with spatial discretizations that are strongly stable, under forward Euler time integration, for nonlinear hyperbolic PDEs with discontinuous solutions. The numerical examples in Sect. 1.2 demonstrate that the theoretical advantage of these methods provides a significant benefit in practice. The SSP theory has benefited from the recent discovery of its close relation to the theory of contractivity (see Sect. 1.4). This connection has allowed a more complete and efficient study of SSP methods with optimal SSP coefficient. Furthermore, contractivity theory allows us to conclude that the SSP coefficient is not only sufficient but necessary for strong stability preservation in an arbitrary norm for an arbitrary semi-discretization that satisfies a strong stability condition under forward Euler integration. Thanks in large part to these theoretical advances, optimal SSP methods of multistep and Runge–Kutta type have been thoroughly investigated, and their development seems to be essentially complete.

In this paper we reviewed the state-of-the-art of SSP methods, and presented explicit and implicit Runge–Kutta and multistep methods, and explicit spectral deferred correction methods, as well as a bound on the SSP coefficient of all explicit general linear methods. To date, low storage Runge–Kutta methods with extra stages [45, 72, 82] and the multistep methods (with special starting conditions) of [73] have emerged as the most promising explicit SSP methods in terms of allowable time-step and computational efficiency. Future work should include further testing of these relatively new methods, in combination with different spatial

discretizations on a wide range of problems (for example, see [52] for a study of various SSP Runge–Kutta methods combined with discontinuous Galerkin methods).

The SSP Runge–Kutta methods tend to have a variety of nice properties, such as small error constants and large regions of absolute stability. The explicit methods have efficient low storage implementations, and the implicit methods are singly diagonally implicit or diagonally implicit. Furthermore, they have provable existence and uniqueness properties.

It has been demonstrated that implicit SSP methods are unlikely to be efficient enough to out-perform the explicit methods. The very restrictive bound $c_{\text{eff}} \leq 2$ has been proved for multistep methods [40, 57] and conjectured for Runge–Kutta methods [47]. Unconditionally SSP methods have been found by looking beyond the class of general linear methods; however, these have proven to reduce to first order for the step-sizes of interest [62].

A promising area of future work is the study of explicit SSP general linear methods. Efficient SSP methods of order greater than four are frequently desirable, particularly when dealing with high order spatial discretizations. All explicit fifth order SSP Runge–Kutta methods (see [72]) require the use of downwind-biased operators, which is sometimes inconvenient; high order SSP multistep methods have large memory requirements, and the more efficient higher order methods with special starting procedures sometimes generate oscillation which may not be acceptable when the SSP property is sought [73]. The higher order implicit methods developed in [47] do not have a very large step-size and so are costly to implement and may not be desirable for most applications. It is hoped that better high order methods may be found by investigating the larger class of general linear methods, even considering the bound on the step-size presented in Sect. 2.3. For a first effort in this direction, see [37].

Other promising areas of research include implicit general linear SSP methods, implicit-explicit SSP methods, and the study of the SSP properties of other widely used numerical methods. Implicit SSP general linear methods may provide an advantage over Runge–Kutta or multistep methods by allowing a larger step-size [47, 57], while the development of implicit-explicit methods whose explicit component is SSP [34, 39] will be helpful in problems in which the step-size is restricted by the stiff component while the SSP condition is needed for the convective component. For some types of time integration methods that are often applied to hyperbolic PDEs, the SSP property has not been analyzed. We have reviewed the recent analysis for the class of explicit spectral deferred correction methods in Sect. 5. Other types of methods for which SSP results would be helpful include other types of deferred correction methods, extrapolation methods, and exponential time differencing methods.

References

1. Baiotti, L., Hawke, I., Montero, P.J., Loffler, F., Rezzolla, L., Stergioulas, N., Font, J.A., Seidel, E.: Three-dimensional relativistic simulations of rotating neutron-star collapse to a Kerr black hole. Phys. Rev. D **71** (2005)
2. Balbás, J., Tadmor, E.: A central differencing simulation of the Orszag-Tang vortex system. IEEE Trans. Plasma Sci. **33**(2), 470–471 (2005)
3. Bassano, E.: Numerical simulation of thermo-solutal-capillary migration of a dissolving drop in a cavity. Int. J. Numer. Methods Fluids **41**, 765–788 (2003)
4. Butcher, J.C.: On the implementation of implicit Runge-Kutta methods. BIT **6**, 237–240 (1976)
5. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations. Wiley, New York (2003)
6. Caiden, R., Fedkiw, R.P., Anderson, C.: A numerical method for two-phase flow consisting of separate compressible and incompressible regions. J. Comput. Phys. **166**, 1–27 (2001)

7. Carrillo, J., Gamba, I.M., Majorana, A., Shu, C.-W.: A WENO-solver for the transients of Boltzmann-Poisson system for semiconductor devices: performance and comparisons with Monte Carlo methods. J. Comput. Phys. **184**, 498–525 (2003)
8. Chen, M.-H., Cockburn, B., Reitich, F.: High-order RKDG methods for computational electromagnetics. J. Sci. Comput. **22–23**, 205–226 (2005)
9. Cheng, L.-T., Liu, H., Osher, S.: Computational high-frequency wave propagation using the level set method, with applications to the semi-classical limit of Schrodinger equations. Commun. Math. Sci. **1**(3), 593–621 (2003)
10. Cheruvu, V., Nair, R.D., Tufo, H.M.: A spectral finite volume transport scheme on the cubed-sphere. Appl. Numer. Math. **57**, 1021–1032 (2007)
11. Cockburn, B., Li, F., Shu, C.-W.: Locally divergence-free discontinuous Galerkin methods for the Maxwell equations. J. Comput. Phys. **194**, 588–610 (2004)
12. Cockburn, B., Qian, J., Reitich, F., Wang, J.: An accurate spectral/discontinuous finite-element formulation of a phase-space-based level set approach to geometrical optics. J. Comput. Phys. **208**, 175–195 (2005)
13. Cockburn, B., Shu, C.-W.: TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. Math. Comput. **52**, 411–435 (1989)
14. Dahlquist, G., Jeltsch, R.: Generalized disks of contractivity for explicit and implicit Runge–Kutta methods. Technical Report, Department of Numerical Analysis and Computational Science, Royal Institute of Technology, Stockholm (1979)
15. Dekker, K., Verwer, J.G.: Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations. CWI Monographs, vol. 2. North-Holland, Amsterdam (1984)
16. Del Zanna, L., Bucciantini, N.: An efficient shock-capturing central-type scheme for multidimensional relativistic flows: I. hydrodynamics. Astron. Astrophys. **390**, 1177–1186 (2002)
17. Dutt, A., Greengard, L., Rokhlin, V.: Spectral deferred correction methods for ordinary differential equations. BIT **40**, 241–266 (2000)
18. Enright, D., Fedkiw, R., Ferziger, J., Mitchell, I.: A hybrid particle level set method for improved interface capturing. J. Comput. Phys. **183**, 83–116 (2002)
19. Feng, L.-L., Shu, C.-W., Zhang, M.: A hybrid cosmological hydrodynamic/n-body code based on a weighted essentially nonoscillatory scheme. Astrophys. J. **612**, 1–13 (2004)
20. Ferracina, L., Spijker, M.N.: Stepsize restrictions for the total-variation-diminishing property in general Runge-Kutta methods. SIAM J. Numer. Anal. **42**, 1073–1093 (2004)
21. Ferracina, L., Spijker, M.N.: Computing optimal monotonicity-preserving Runge-Kutta methods. Technical Report MI2005-07, Mathematical Institute, Leiden University (2005)
22. Ferracina, L., Spijker, M.N.: An extension and analysis of the Shu-Osher representation of Runge-Kutta methods. Math. Comput. **249**, 201–219 (2005)
23. Ferracina, L., Spijker, M.N.: Strong stability of singly-diagonally-implicit Runge-Kutta methods. Appl. Numer. Math. (2008). doi:10.1016/j.apnum.2007.10.004
24. Gottlieb, D., Tadmor, E.: The CFL condition for spectral approximations to hyperbolic initial-boundary value problems. Math. Comput. **56**, 565–588 (1991)
25. Gottlieb, S.: On high order strong stability preserving Runge-Kutta and multi step time discretizations. J. Sci. Comput. **25**, 105–127 (2005)
26. Gottlieb, S., Gottlieb, L.J.: Strong stability preserving properties of Runge-Kutta time discretization methods for linear constant coefficient operators. J. Sci. Comput. **18**, 83–109 (2003)
27. Gottlieb, S., Ruuth, S.J.: Optimal strong-stability-preserving time-stepping schemes with fast downwind spatial discretizations. J. Sci. Comput. **27**, 289–303 (2006)
28. Gottlieb, S., Shu, C.-W.: Total variation diminishing Runge-Kutta schemes. Math. Comput. **67**, 73–85 (1998)
29. Gottlieb, S., Shu, C.-W., Tadmor, E.: Strong stability preserving high-order time discretization methods. SIAM Rev. **43**, 89–112 (2001)
30. Hairer, E., Wanner, G.: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Springer Series in Computational Mathematics, vol. 14. Springer, Berlin (1991)
31. Harten, A.: High resolution schemes for hyperbolic conservation laws. J. Comput. Phys. **49**, 357–393 (1983)
32. Higueras, I.: On strong stability preserving time discretization methods. J. Sci. Comput. **21**, 193–223 (2004)
33. Higueras, I.: Representations of Runge-Kutta methods and strong stability preserving methods. SIAM J. Numer. Anal. **43**, 924–948 (2005)
34. Higueras, I.: Strong stability for additive Runge-Kutta methods. SIAM J. Numer. Anal. **44**, 1735–1758 (2006)
35. Horvath, Z.: On the positivity of matrix-vector products. Linear Algebra Appl. **393**, 253–258 (2004)

36. Horvath, Z.: On the positivity step size threshold of Runge-Kutta methods. Appl. Numer. Math. **53**, 341–356 (2005)
37. Huang, C.: Strong stability preserving hybrid methods. Appl. Numer. Math. (2008). doi: 10.1016/j.apnum.2008.03.030
38. Hundsdorfer, W., Ruuth, S.J.: On monotonicity and boundedness properties of linear multistep methods. Math. Comput. **75**(254), 655–672 (2005)
39. Hundsdorfer, W., Ruuth, S.J.: IMEX extensions of linear multistep methods with general monotonicity and boundedness properties. J. Comput. Phys. **225**, 201–2042 (2007)
40. Hundsdorfer, W., Ruuth, S.J., Spiteri, R.J.: Monotonicity-preserving linear multistep methods. SIAM J. Numer. Anal. **41**, 605–623 (2003)
41. Jeltsch, R., Nevanlinna, O.: Stability of explicit time discretizations for solving initial value problems. Numer. Math. **37**, 61–91 (1981)
42. Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**, 202–228 (1996)
43. Jin, S., Liu, H., Osher, S., Tsai, Y.-H.R.: Computing multivalued physical observables for the semiclassical limit of the Schrodinger equation. J. Comput. Phys. **205**, 222–241 (2005)
44. Kennedy, C.A., Carpenter, M.H., Lewis, R.M.: Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. Appl. Numer. Math. **35**, 177–219 (2000)
45. Ketcheson, D.I.: Highly efficient strong stability preserving Runge-Kutta methods with low-storage implementations. SIAM J. Sci. Comput. (2008, to appear)
46. Ketcheson, D.I.: Computation of optimal contractive general linear methods and strong stability preserving linear multistep methods. Math. Comput. (to appear)
47. Ketcheson, D.I., Macdonald, C.B., Gottlieb, S.: Optimal implicit strong stability preserving Runge-Kutta methods. Appl. Numer. Math. (2008). doi: 10.1016/j.apnum.2008.03.034
48. Ketcheson, D.I., Macdonald, C.B., Gottlieb, S.: Numerically optimal SSP Runge–Kutta methods (website). http://www.cfm.brown.edu/people/sg/ssp.html (2007)
49. Ketcheson, D.I., Robinson, A.C.: On the practical importance of the SSP property for Runge-Kutta time integrators for some common Godunov-type schemes. Int. J. Numer. Methods Fluids **48**, 271–303 (2005)
50. Kraaijevanger, J.F.B.M.: Absolute monotonicity of polynomials occurring in the numerical solution of initial value problems. Numer. Math. **48**, 303–322 (1986)
51. Kraaijevanger, J.F.B.M.: Contractivity of Runge-Kutta methods. BIT **31**, 482–528 (1991)
52. Kubatko, E.J., Westerink, J.J., Dawson, C.: Semi discrete discontinuous Galerkin methods and stage-exceeding-order, strong-stability-preserving Runge-Kutta time discretizations. J. Comput. Phys. **222**, 832–848 (2007)
53. Kurganov, A., Tadmor, E.: New high-resolution schemes for nonlinear conservation laws and convection-diffusion equations. J. Comput. Phys. **160**, 241–282 (2000)
54. Labrunie, S., Carrillo, J., Bertrand, P.: Numerical study on hydrodynamic and quasi-neutral approximations for collisionless two-species plasmas. J. Comput. Phys. **200**, 267–298 (2004)
55. Lax, P.D.: Gibbs Phenomena. J. Sci. Comput. **28**, 445–449 (2006)
56. Lenferink, H.W.J.: Contractivity-preserving explicit linear multistep methods. Numer. Math. **55**, 213–223 (1989)
57. Lenferink, H.W.J.: Contractivity-preserving implicit linear multistep methods. Math. Comput. **56**, 177–199 (1991)
58. Levy, D., Tadmor, E.: From semi-discrete to fully discrete: stability of Runge-Kutta schemes by the energy method. SIAM Rev. **40**, 40–73 (1998)
59. Liu, X.-D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. J. Comput. Phys. **115**(1), 200–212 (1994)
60. Liu, Y., Shu, C.-W., Zhang, M.: Strong stability preserving property of the deferred correction time discretization. J. Comput. Math. **26**, 633–656 (2008)
61. Macdonald, C.B.: Constructing high-order Runge-Kutta methods with embedded strong-stability-preserving pairs. Master's Thesis, Simon Fraser University (2003)
62. Macdonald, C.B., Gottlieb, S., Ruuth, S.: A numerical study of diagonally split Runge-Kutta methods for PDEs with discontinuities (2008, submitted)
63. Majda, A., Osher, S.: A systematic approach for correcting nonlinear instabilities. the Lax-Wendroff scheme for scalar conservation laws. Numer. Math. **30**, 429–452 (1978)
64. Mignone, A.: The dynamics of radiative shock waves: linear and nonlinear evolution. Astrophys. J. **626**, 373–388 (2005)
65. Minion, M.L.: Semi-implicit spectral deferred correction methods for ordinary differential equations. SIAM J. Numer. Anal. **41**, 605–623 (2003)
66. Nessyahu, H., Tadmor, E.: Non-oscillatory central differencing for hyperbolic conservation laws. J. Comput. Phys. **87**, 408–463 (1990)

67. Osher, S., Chakravarthy, S.: High resolution schemes and the entropy condition. SIAM J. Numer. Anal. **21**, 955–984 (1984)
68. Osher, S., Tadmor, E.: On the convergence of difference approximations to scalar conservation laws. Math. Comput. **50**, 19–51 (1988)
69. Pantano, C., Deiterding, R., Hill, D.J., Pullin, D.I.: A low numerical dissipation patch-based adaptive mesh refinement method for large-eddy simulation of compressible flows. J. Comput. Phys. **221**, 63–87 (2007)
70. Patel, S., Drikakis, D.: Effects of preconditioning on the accuracy and efficiency of incompressible flows. Int. J. Numer. Methods Fluids **47**, 963–970 (2005)
71. Peng, D., Merriman, B., Osher, S., Zhao, H., Kang, M.: A PDE-based fast local level set method. J. Comput. Phys. **155**, 410–438 (1999)
72. Ruuth, S.J.: Global optimization of explicit strong-stability-preserving Runge-Kutta methods. Math. Comput. **75**, 183–207 (2006)
73. Ruuth, S.J., Hundsdorfer, W.: High-order linear multistep methods with general monotonicity and boundedness properties. J. Comput. Phys. **209**, 226–248 (2005)
74. Ruuth, S.J., Spiteri, R.J.: Two barriers on strong-stability-preserving time discretization methods. J. Sci. Comput. **17**, 211–220 (2002)
75. Ruuth, S.J., Spiteri, R.J.: High-order strong-stability-preserving Runge-Kutta methods with downwind-biased spatial discretizations. SIAM J. Numer. Anal. **42**, 974–996 (2004)
76. Sahinidis, N.V., Tawarmalani, M.: BARON 7.2: Global optimization of mixed-integer nonlinear programs, user's manual. Available at http://www.gams.com/dd/docs/solvers/baron.pdf (2004)
77. Shu, C.-W.: Total-variation diminishing time discretizations. SIAM J. Sci. Statist. Comput. **9**, 1073–1084 (1988)
78. Shu, C.-W.: A survey of strong stability-preserving high-order time discretization methods. In: Collected Lectures on the Preservation of Stability under Discretization. SIAM, Philadelphia (2002)
79. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
80. Spijker, M.N.: Contractivity in the numerical solution of initial value problems. Numer. Math. **42**, 271–290 (1983)
81. Spijker, M.N.: Stepsize conditions for general monotonicity in numerical initial value problems. SIAM J. Numer. Anal. **45**, 1226–1245 (2007)
82. Spiteri, R.J., Ruuth, S.J.: A new class of optimal high-order strong-stability-preserving time discretization methods. SIAM J. Numer. Anal. **40**, 469–491 (2002)
83. Spiteri, R.J., Ruuth, S.J.: Nonlinear evolution using optimal fourth-order strong-stability-preserving Runge-Kutta methods. Math. Comput. Simul. **62**, 125–135 (2003)
84. Strang, G.: Accurate partial difference methods II: nonlinear problems. Numer. Math. **6**, 37–46 (1964)
85. Strikwerda, J.C.: Finite Difference Schemes and Partial Differential Equations. Cole Mathematics Series. Wadsworth and Brooks, California (1989)
86. Sun, Y., Wang, Z.J., Liu, Y.: Spectral (finite) volume method for conservation laws on unstructured grids VI: Extension to viscous flow. J. Comput. Phys. **215**, 41–58 (2006)
87. Sweby, P.K.: High resolution schemes using flux limiters for hyperbolic conservation laws. SIAM J. Numer. Anal. **21**, 995–1011 (1984)
88. Tadmor, E.: Approximate solutions of nonlinear conservation laws. In: Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lectures Notes from CIME Course Cetraro, Italy, 1997. Lecture Notes in Mathematics, vol. 1697, pp. 1–150. Springer, Berlin (1998)
89. Tanguay, M., Colonius, T.: Progress in modeling and simulation of shock wave lithotripsy (swl). In: Fifth International Symposium on Cavitation (CAV2003), number OS-2-1-010 (2003)
90. van de Griend, J.A., Kraaijevanger, J.F.B.M.: Absolute monotonicity of rational functions occurring in the numerical solution of initial value problems. Numer. Math. **49**, 413–424 (1986)
91. Wang, R., Spiteri, R.J.: Linear instability of the fifth-order WENO method. SIAM J. Numer. Anal. **45**(5), 1871–1901 (2007)
92. Wang, Z.J., Liu, Y.: The spectral difference method for the 2D Euler equations on unstructured grids. In: 17th AIAA Computational Fluid Dynamics Conference. AIAA, Washington (2005)
93. Wang, Z.J., Liu, Y., May, G., Jameson, A.: Spectral difference method for unstructured grids II: Extension to the Euler equations. J. Sci. Comput. **32**(1), 45–71 (2007)
94. Williamson, J.H.: Low-storage Runge-Kutta schemes. J. Comput. Phys. **35**, 48–56 (1980)
95. Xia, Y., Xu, Y., Shu, C.-W.: Efficient time discretization for local discontinuous Galerkin methods. Discrete Continuous Dyn. Syst. Ser B **8**, 677–693 (2007)
96. Zhang, W., MacFayden, A.I.: RAM: A relativistic adaptive mesh refinement hydrodynamics code. Astrophys. J. Suppl. Ser. **164**, 255–279 (2006)